Object Identification for Ubiquitous Networking

Gyu Myoung Lee[1], Jun Kyun Choi[2], Noel Crespi[1]
[1]Institut TELECOM SudParis
9 rue Charles Fourier, 91011, Evry France
{gm.lee, noel.crespi}@it-sudparis.eu
[2]Information and Communications University (ICU)
119 Munjidong Youseonggu Daejeon, 305-732, Korea
jkchoi@icu.ac.kr

Abstract — **We explain the concept of ubiquitous networking including object to object communications and specify naming and addressing issues for object identification. In order to use Host Identity Protocol (HIP) for connecting objects in ubiquitous networking environment, we propose the extensions of HIP according to mapping relationships between host and objects. In addition, we provide packet formats and considerations for HIP extensions concerning objects. Our proposal can be used for location management, networked-ID services, etc.**

Keywords — **Ubiquitous Networking, Object, HIP.**

1. Introduction

The one of new capabilities for future network will be the ubiquitous networking. The ubiquitous networking means networking capabilities to support various classes of applications/services which require "Any Services, Any Time, Any Where and Any Devices" operation. This networking capability should support human-to-human, human-to-objects (e.g., devices and/or machines) and objects-to-objects communications [1, 2]. So, the ubiquitous networking enables objects to communicate and access services without any restrictions at services, places, time, users, etc.

The goal of ubiquitous networking is eventually to provide capabilities for connecting all of objects in future network. To support this capability, the current network should be extended in the aspects of functional capabilities, especially end devices aspects. One of ultimate objectives of ubiquitous networking is to meet the challenge of seamless connection of anything (e.g., humans and objects) in Internet infrastructure consisting of devices, network, platform, and contents. Business areas using ubiquitous networking in future network will extend to all of industries such as education, commerce, finance, logistics, transportation, circumstances, emergency service, agriculture, medical services, etc. Thus the emerging ubiquitous networking will impact on the whole industries including network industry.

There are so many kinds of devices to be supported for ubiquitous networking in future network. Radio Frequency Identifier (RFID) tag, sensors, smart cards, medical devices, navigation devices as well as the existing personal devices such as Personal Computer (PC), Personal Digital Assistance (PDA), etc will be connected to ubiquitous networks. Accordingly, for ubiquitous networking, IP connectivity will be one of very vital features. IP connectivity is very useful when there is the need for objects to communicate with each other within a network, and when objects have to be reachable from outside their networks. This paper 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, in particular object to object communications will be provided using the new concept of end points considering object. Therefore, in this paper, we focus on how to support ubiquitous networking using the extensions of existing Host Identity Protocol (HIP).

The role of 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 [3, 4].

In ubiquitous networking environment, 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 in current network.

For identification of network entities, we consider new type of identifiers (e.g., RFID code [5], content ID [6], etc) for object and describe specific requirements for object identification in naming and addressing point of view. In order to use HIP for ubiquitous networking including object to object communication, we propose the extensions of HIP according to mapping relationships between host and object(s). In addition, we provide packet formats and considerations for HIP extensions.

The remainder of the paper is organized as follows. In Section 2, we classify network entities to be identified in the network and show examples of identification codes for objects. In Section 3, we discuss mapping relationship between identities in layered architecture and present two kinds of one to many mapping. Then, in Section 4, we propose the extensions of HIP for object identification in order to support ubiquitous networking and present the related packet formats. We also provide the protocol procedure and identify several considerations. Finally in Section 5, we come to a conclusion.

2. Identification codes for objects

2.1. Classification of network entities to be identified

The following 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.

2.2. 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 [7].

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
  - Object ID: the identifier for identifying a physical entity which is attached in the network at application layer. (e.g., RFID, content ID, etc)
  - Others: the logical number or the identifier which is required for service provision. (e.g., telephone number, URL/URI, etc)
- Communication IDs: the physical identifier or the logical identifier for IP connectivity in the network in order to provide communication capabilities. (e.g., session/ protocol ID, IP address, MAC address, etc)

2.3. Examples of object ID

Here we introduce Electronic Product Code (EPC) for RFID/sensor and content ID as examples of object ID which is used for identification in the network.

An EPC is very important in ubiquitous networking environment. The 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 [5].

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 [6].

3. Naming and addressing using object identification

3.1. Layered architecture for naming and addressing

The vertical 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 identity processing such as mapping/binding [8].

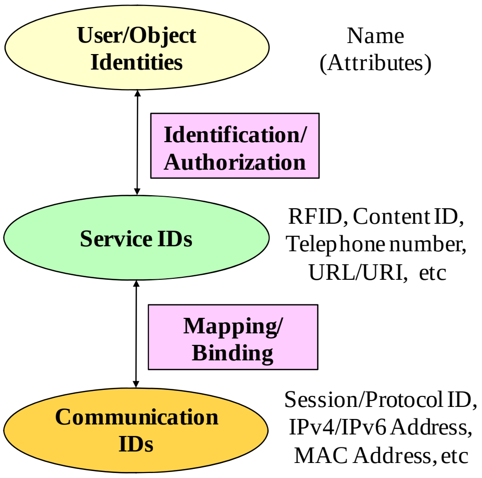
User/Object Identities
Name (Attributes)
Identification/ Authorization
Service IDs
RFID, Content ID, Telephone number, URL/URI, etc
Mapping/ Binding
Communication IDs
Session/Protocol ID, IPv4/IPv6 Address, MAC Address, etc

Figure 1. Identity processing in vertical layered architecture

The architecture in Figure 2 has the following relationship at each layer [9].

An application is an entity that is either an instance of a specific application service or a specific data object. The identity of the object persists over time and is not tied to the end system hosting the service or data.

An end point (e.g., host entity) resides in a node in a network. Hosts may be part of multiple networks at the same time. The identity of the host entity is independent of its current location(s) in the network. The host's locations are determined by the locators of the points of network attachment which it attaches to with specific communication interfaces. Host entities attach to the network at points of network attachment, which also define generic locations in the network topology.

A Location is identified with some sort of network address or locator. These locators often depend on the network topology and technology used.

A Path represents a physical route between physical devices (e.g., network attachments)

There are two connectivity abstractions. A bearer, the upper-level connectivity abstraction, connects two points of application attachment with one another. A flow, the lower-level connectivity abstraction, runs between two points of network attachment.

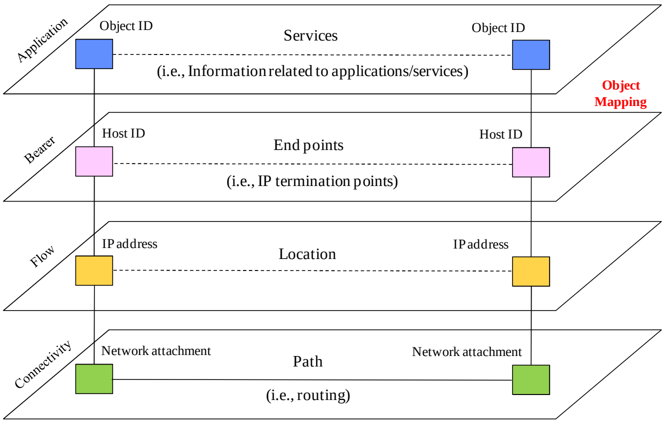

Figure 2. Layered mapping architecture between identities

One purpose of defining a layered architecture is to provide mappings/bindings between entities at different levels. With mappings/bindings, identities of entities become location independent. We can provide different types of mobility, such as for nodes and services.

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.

– Protection of object (including rights management)
– Connecting to anything using object identification
– Service and location discovery

Therefore, based on mapping relationships, we propose an identity protocol for objects, i.e., HIP extensions, in order to perform mapping/binding capability and support the features required in communications between objects.

3.2. Mapping relationships for object identification

For object identification, we can consider the following relationships between host and object(s):

First, 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 a host can be allocated a telephone number as service ID and be treated the same object.

Second, 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. We can consider two kinds of one to many mapping 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.

Indirection mapping can additionally support advanced mobility schemes, such as moving objects, and explicitly control middleboxes. Indirection means that the name of an entity does not bind an ID at the lower layer within the same node, but sideways to another location where an intermediary takes care of forwarding the communication to the entity's actual location. A simple application of this mechanism enables servers to operate behind a gateway without explicit configuration.

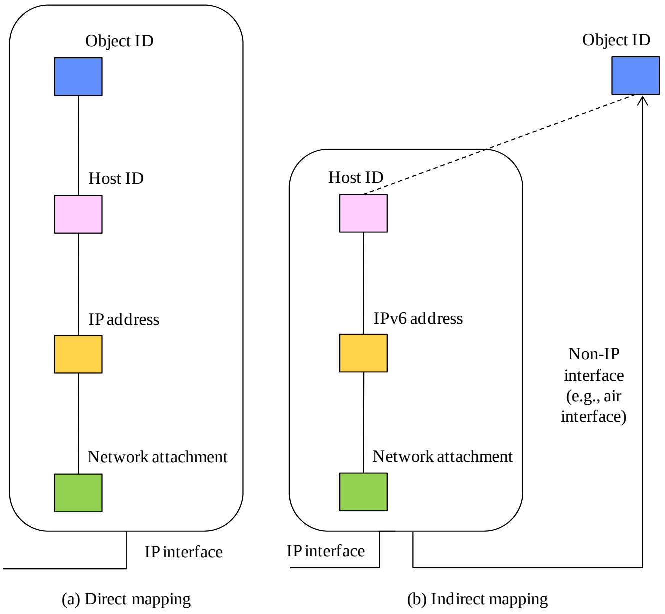

Figure 3. Mapping between host and object(s)

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

- Perform routing using network prefix information and identification code using service Ids together
- Provide the connectivity to end device without additional equipment such as Network Address Translator (NAT)
- Scalability with new name space on top of network layer for supporting objects-to-objects communications
- Security support using HIP hash function, etc

4. HIP extensions for object identification

4.1. HIP extensions

In order to support ubiquitous networking, we propose object identification using HIP extensions. According to the mapping relationships as shown in Figure 3, we extend the current HIP for objects as follow:

- **Case #1: Objects in a host (direct mapping)**
  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, we propose to newly define a new TLV[1], i.e., Object_ID (see Section 4.2.).
- **Case #2: Remote objects (indirect mapping)**
  In 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, we propose to newly specify Object Identity Tag (OIT) in HIP message. Each OIT typically identifies a service and can also identify end point.

4.2. Packet formats

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

- HIP header (include OIT)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   | Header Length |0| Packet Type |  VER. | RES.|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Checksum             |           Controls            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Sender's Host/Object Identity Tag (HIT/OIT)          |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Receiver's Host/Object Identity Tag (HIT/OIT)         |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
/                        HIP Parameters                         /
/                                                               /
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

- Object_ID (newly defined from HOST_ID of HIP)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           OI Length           |DI-type|      DI Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Object Identity                        /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |      Domain Identifier        /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                       |       Padding         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type	to be defined (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[2] or NAI[3] 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 [4]. Other packet formats are subject to change according to HIP. In the proposed extensions of HIP, we need to find the solution for security association with object identity.

For security association, there is an alternative to keep the existing HOST_ID and add new Domain Identifier type for the object ID [10]. In this case, we can use the existing HIP for security association. For this method, we need further discussion.

4.3. Protocol procedure for connecting objects

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 directly connect to object(s) using routing.

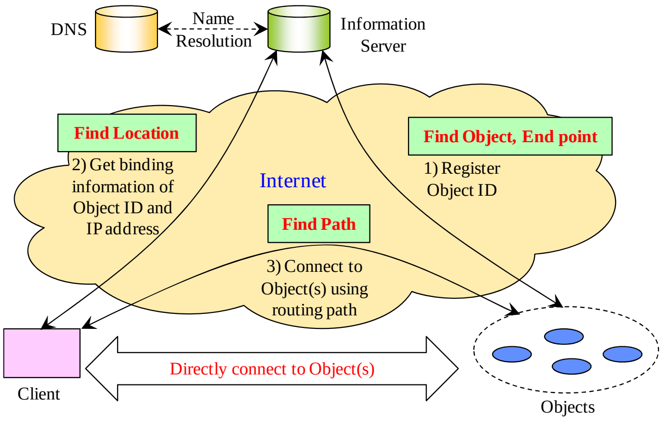

Figure 4. Protocol procedure for connecting objects

[1] TLV: Type, Length, Value

[2] FQDN: Fully Qualified Domain Name
[3] NAI: Network Access Identifier

4.4. Considerations for HIP Extensions

For HIP extensions, we need to further consider the following:

- Security association for secure binding between object identity and host identity [11].
- 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 OITs
- Protocol overhead: real time communications and some limitation of power and packet size, lightweight identity handshake for datagram transactions is critical.
- 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.
- 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

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

5. Conclusion

This paper has explained the concept of ubiquitous networking including object to object communications and specified naming and addressing issues for object identification. In order to use HIP for ubiquitous networking, we have proposed the extended architecture and mechanism of HIP according to mapping relationships between host and object(s). In addition, we have provided packet formats and considerations for HIP extensions concerning objects. The proposed method can provide the connectivity to all of objects using object identification in ubiquitous networking environment and be used for location management, networked-ID services, etc.

References

[1] Gyu Myoung Lee, Jun Kyun Choi, and 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] ITU-T TD280Rev1 (NGN-GSI), “Initial Draft Recommendation Y.NGN-UbiNet, “Overview and Principles for Ubiquitous Networking in NGN””, *work in progress*, September 2008.

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

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

[5] EPCglobal, “EPCglobal Object Name Service (ONS) 1.0.1”, May 2008.

[6] Content ID Forum (cIDf), “cIDf Specification 2.0”, April 2007.

[7] Gyu Myoung Lee, Jun Kyun Choi, and Taesoo Chung, “Address structure for supporting ubiquitous networking using IPv6,” the 10th International Conference on Advanced Communication Technology, pp. 1088~1090, February 2008.

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

[9] Bengt Ahlgren, Lars Eggert, Borje Ohlman, Jarno Rajahalme, and Andreas Schieder, “Names, addresses and identities in ambient networks,” *International conference on mobile computing and networking*, pp.33~37, 2005.

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

[11] Heer, Varjonen, “HIP Certificates,” *IETF Internet-Draft*, draft-ietf-hip-cert-00.txt, *work in progress*, October 2008.