6lo
Internet-Draft
Intended status: Standards Track
Expires: February 15, 2015

G. Rizzo, Ed.
AJ. Jara, Ed.
A. Olivieri
Y. Bocchi
HES-SO
MR. Palattella
SnT/Univ. of Luxembourg
L. Ladid
SnT/Univ. of Luxembourg/IPv6 Forum

August 14, 2014

IPv6 mapping to non-IP protocols
draft-rizzo-6lo-6legacy-01

Abstract

IPv6 is an important enabler of the Internet of Things, since it provides an addressing space large enough to encompass a vast and ubiquitous set of sensors and devices, allowing them to interconnect and interact seamlessly. To date, an important fraction of those devices is based on networking technologies other than IP. An important problem to solve in order to include them into an IPv6-based Internet of Things, is to define a mechanism for assigning an IPv6 address to each of them, in a way which avoids conflicts and protocol aliasing.

The only existing proposal for such a mapping leaves many problems unsolved and it is nowadays inadequate to cope with the new scenarios which the Internet of Things presents. This document defines a mechanism, 6TONon-IP, for assigning automatically an IPv6 address to devices which do not support IPv6 or IPv4, in a way which minimizes the chances of address conflicts, and of frequent configuration changes due to instability of connection among devices. Such a mapping mechanism enables stateless autoconfiguration for legacy technology devices, allowing them to interconnect through the Internet and to fully integrate into a world wide scale, IPv6-based IoT.

Status of This Memo

Copyright Notice

Table of Contents

1. Terminology

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 [RFC2119].

2. Introduction

The Future Internet and the IPv6 protocol enable a new generation of techniques for accessing the network, which extend the Internet seamlessly to personal devices, sensors, home appliances, enabling the so called ''Internet of Things'' (IoT). One of the key issues which presently hampers the development of IoT and limits its potential is the lack of an efficient common framework for the integration among the vast and diverse set of protocols and technologies which compose it. Current sensors and their application environments employ a large set of technologies which lack efficient interoperability. Some associations of manufacturers have been formed to build a common technological framework in specific application domains, e.g. KNX for building automation (http://www.knx.org/), ZigBee (ZigBee Alliance) (http://www.zigbee.org/), and protocols such as X10 and CAN. Such frameworks are based on very different architectures, and the protocols which compose them are generally not interoperable. Finally, most of these technologies were designed in a context of small and local networks, with limited capabilities, and they were not conceived for integration within the Internet. One of the ideas at the basis of the IoT is the constitution of a common set of protocols which enables the interaction between devices through the Internet. By enabling interaction through the Internet, new services could be conceived and implemented, increasing the value produced by the IoT infrastructure. The adoption of a common framework may make more economically convenient its deployment, and foster the development of new smart environments (buildings, cities, etc), ultimately making possible the full realization of the potential of the IoT. As deployment of new sensors is typically expensive, it is unthinkable of putting to disuse an installed set of sensors, once a new set of devices (typically, IPv6 enabled) is deployed. This is not an uncommon case, as the set of deployed legacy devices (sensors, actuators) is to date very large. Rather, mechanisms are needed to integrate legacy devices into a common IoT platform, in order to include them in all the present and future services (e.g. devices and services directory, localization services, etc) which will be implemented on the IoT. For these reasons, many designers of the Internet of Things are focusing on building such common access and communications framework. All the proposals (e.g. CoAP, RESTful Web services) presently under discussion are based on IPv6. This has important implications on the addressing of the devices. Indeed a common addressing at the device level is mandatory, in order to implement true Machine to Machine (M2M) communications without Portal Servers, which would make the whole system difficult to integrate and scale. The present document focuses on the network layer aspects of such IPv6 based integration. At the network layer, a mechanism which assigns an IPv6 address to each device is needed, to solve the addressing problem. In this document, we propose a new mechanism for the users and devices to map the different addressing spaces to a common IPv6 one. Our proposed mechanism solves several issues posed by some of the mappings adopted so far. Such mapping makes it possible for every device from each technology to operate through a common framework based on IPv6 and protocols over IPv6 such as RESTFul WebServices and Constrained Application Protocol (CoAP). For each technology, the proposed mechanism maps technology-specific features to a set of fields defined within the IPv6 address. This allows the location and identification of the devices in a multi-protocol card, or in any gateway or Portal Server.

2.1. Examples

In this subsection, we present two examples which help understanding the importance of adopting a common IPv6 based framework for interaction between things, and the need for legacy devices to be individually addressable through IPv6.

2.1.1. Example 1 - Building automation systems and IoT

The IoT is composed by a very large set of devices, which is poised to grow exponentially in the near future. For this reason, a directory service is needed, which offers the possibility to individuate a specific device or set of devices, with given capabilities or within a given geographical region. Let us assume such directory lists devices with their IPv6 addresses, and their function (say a temperature sensor, or a mobile phone, etc). For instance, let us consider the case of someone willing to build a map of temperatures in a given geographical region. Such directory service would allow retrieving the list of available devices within that region, each with its own IPv6 address. Assume some of those devices are legacy, non IP based temperature sensors and part of a given building automation system. Assume also that such system manages several of those temperature sensors. Even if such system would be reachable via IP, without having those sensors individually listed in the directory and appearing as ’’autonomous’’ things, which can be polled directly, one should resort to techniques for retrieving the temperature reading of those sensors which are specific of that building automation technology. This would make more complex the implementation of such a temperature map.

Instead, by having the building automation system expose each sensor as an IPv6 enabled device, the whole set of temperature sensors would be accessible in a homogeneous way, greatly simplifying the task.

2.1.2. Example 2 - KNX and demand-side management

KNX is a standardized (EN 50090, ISO/IEC 14543), OSI-based network communications protocol for intelligent buildings. Among the devices typically managed through KNX, we find:

- Lighting control systems;
- Heating/ventilation and air conditioning devices;
- Shutter/blind and shading control systems; and
- Energy management and electricity/gas/water metering devices.

KNX devices do not support IP. Therefore, in order to connect a KNX home network to the Internet, a gateway (KNXnet/IP router) is necessary. Other technologies for home automation are available nowadays, in which each smart device (air conditioners, washing machines, etc) supports IPv6. Let us consider a scenario in which an utility company offers an agreement to a fraction of its clients. In exchange for a cut on the energy bill, the utility company gains direct control over some appliances at the premises of the client. In this way, by powering off some of those devices in periods when the production cost of power are very high, the utility company realizes potentially high savings.

In order to implement this, the utility company sends commands to a set of devices under its direct control. For recently installed devices, the utility can assume that they support IPv6, and some application layer protocols such as CoAP. Therefore a command to switch off a device would use the IPv6 address to identify the device, and the application layer protocol to send the actual command. But for KNX devices, the command should have another format: the IPv6 address should be the one of the router bridging the IPv6 and the KNX networks, and upper layers protocols should take care of identifying the specific device inside the KNX home network to whom the command should be sent. Having to format a specific query for each specific home automation protocol adds a level of complexity which translates into higher costs of implementation and maintenance of such a service.

3. Reference System

In this section we describe a reference system where the IPv6 mapping is used. Such a system includes:

1. A set of networks running non-IPv6-compatible technologies, each with one or more hosts connected. Such networks generally use

different OSI layer 3 protocols, or they may adopt a technology which does not have any layer 3 protocol.

2. A proxy, which hosts the IPv6 mapping functionality. Such device is typically connected to each of the legacy protocols networks, and it accesses the Internet via the IPv6 protocol. Such IPv6 addressing proxy performs all the necessary conversions and adaptations between IPv6 and the (local) networking protocol of the legacy technologies, in a way which depends on the specific legacy technology considered. This proxy makes use of the IPv6 mapping mechanism in order to transforms the native addressing to IPv6 Host ID and vice versa in a way that depends on the legacy technology.

Though in what follows we will describe the proposed mapping with reference to such a system, the main ideas behind it are more general, and they apply to settings others than the one of reference presented here.

4. Issues addressed through the 6TONon-IP mapping mechanism

In this section we highlight the main open issues regarding assignment of IPv6 addresses to devices which do not support IPv6 or IPv4, and we describe a set of desirable properties for a mechanism for automatic assignment of IPv6 addresses to such devices, which we name henceforth 6TONon-IP. In Appendix A of RFC 4291, a method is described for creating modified EUI-64 format Interface Identifiers out of links or nodes with IEEE EUI-64 Identifiers, or with IEEE 802 48-bit MACs. Moreover, for technologies having other link layer interface identifier, some possible mapping methods are sketched, leaving for each legacy protocol the possibility to define its own mapping method.

In the present document, we propose a mapping mechanism which enables stateless address autoconfiguration for legacy technologies, and which exploits some protocol specific identifier such as link layer interface identifiers, and the like. The proposed mapping mechanism addresses the following issues:

1. Protocol identification: For the legacy protocols to which the mapping described in RFC 4291 does not apply, a mechanism is needed to map an IPv6 address to the right legacy protocol. This feature is necessary in case of devices which operate as proxy for more than one legacy technology at the same time.

2. Inter protocol aliasing: Without a mechanism for identifying the legacy protocol from the host part of the IPv6 address, address conflicts are possible among devices belonging to different

legacy protocols. For instance, this may happen when the link layer interface identifier is the same for two devices belonging to different technologies. As several legacy technologies are characterized by a small addressing space, address conflicts are not so unlikely.

3. Conflicts between IPv6 mapped legacy technology addresses and addresses derived from (modified or not) EUI-64 format interface identifiers.

4. Intra-protocol aliasing: As several legacy technologies are characterized by a small addressing space, it is not unlikely to have two legacy devices, mapped to IPv6 addresses with the same network ID (for instance, in the case in which they belong to two separate networks of the same technology, both connected to a same proxy), and with a same interface identifier, and mapping therefore to a same IPv6 address.

Moreover, the following is a list of desirable properties for a 6TONon-IP mapping:

1. Consistency: A host should get the same IPv6 address every time it connects to a same legacy network, assuming that the configuration of all the other devices in that network remains unchanged. This allows avoiding to advertise a new address every time the host reconnects. This feature might be particularly important for devices which are not always "on", or which are not permanently connected.

2. Local Uniqueness: For devices which have an IPv6 address with a same network part, the host part should be unique for each host. This property allows avoiding address conflicts.

3. Uniqueness within the whole Internet: Coherently with the IoT vision, the host part of an IPv6 address associated to a host should be unique within the whole Internet.

Depending on the specific legacy protocol, there might be protocol specific limitations to the satisfaction of these properties. In particular, for those protocols which do not have an interface identifier which is unique, properties 1) and 2) cannot be fully satisfied. Indeed, no mapping can solve address conflicts which take place inside a legacy protocol network. When legacy protocols have a interface identifier which is unique, this can be used to produce a unique host part of an IPv6 address, and its uniqueness would guarantee the satisfaction of properties 1), 2) and 3).

5. 6TONon-IP Mapping Method

In this section we describe the proposed strategy for forming IPv6 addresses from legacy protocol information, and the address format that derives from it. We assume that (one or more) 64 bits Network ID prefixes are given to the mapping function, which therefore computes the 64 bits of the Host ID part of the address (IPv6 interface identifier), in order to form a full IPv6 address.

The input of the proposed mapping function consists in the interface identifier of the legacy protocol.

In the proposed mapping method, the resulting Host ID part (IPv6 interface identifier) is composed by six fields, as shown in Figure 1:

- A Technology ID field (11 bits), containing a code which identifies the specific legacy protocol. This field is split into two parts, one of 6 bits, and another of 5 bits.

- U/L bit (1 bit), in order to keep compatibility with the mapping EUI-64 [RFC4291]. The U/L bit is the seventh bit of the first byte and is used to determine whether the address is universally or locally administered. This bit is set to "0", in order to indicate local scope, analogously to what proposed in [RFC4291]. This choice prevents address conflicts with IPv6 interface identifier generated from IEEE EUI-64 identifiers or IEEE 48-bit MAC identifiers.

- A Reserved field (4 bits). This field can be used in the future for the identification of different interfaces for a same technology (in the same subnetwork).

- Technology Mapping field (32 bits), which maps the interface identifier of the legacy protocol. For those protocols for which the IID is not larger than 32 bits, this field contains the 32 bits of the IID. For IID which are larger than 32 bits, a hashing function is used instead of direct mapping. In particular, some hashing algorithms such as CRC-32 are suggested. Hashing satisfies the requirements of consistency and uniqueness within a subnet with a very high probability, which depends on the hashing algorithm used. This field is split into two parts, one of 8 bits, and another of 24 bits.

- The fourth and fifth bytes are both set to to "0x00", in order not to conflict with EUI-64 interface identifiers.

The resulting format of the Host ID part of the IPv6 address obtained from the mapping is indicated in Figure 1.

```
+--------+-------+------+--------+--------+----------+---------+
| Tech.  |  U/L  | Tech. | Reserved| Tech.   |  EUI-64  | Tech.   |
|  ID    |  "0"  |  ID   |         | Mapping| "0x0000" | Mapping |
|  MSB   |       |  LSB  |         |  MSB   |          |   LSBs  |
|(6 bits)|(1 bit)|(5 bit)| (4 bits)|(8 bits)| (16 bits)|(24 bits)|
+--------+-------+------+--------+--------+----------+---------+
```

Figure 1: general format of the host ID part for legacy protocols

6. Examples

In this section we illustrate the proposed mapping method by applying it on some examples.

6.1. Example 1 - EIB/KNX

We assume the legacy protocol is EIB/KNX. This device has two kind of addresses: On the one hand, a logical address for management of group operations, and on the other hand, an individual address for identification of the device in the topology.

The mapping will be focused for the individual address. This includes an Area ID (4 bits), Line ID (8 bits), and Device ID (8 Bits). An example, is the value 0x1/0x01/0x01 for a sensor connected in the Area ID 0x1, Line ID 0x01, and Device ID 0x01.

We apply a hash (CRC-32) to the sequence 0x10101. The result is 0xDEA258A5.

Let us assume that EIB/Konnex Technology ID is "0". Thereby, the IPv6 interface identifier is "0000:DE00:00A2:58A5", considering the documentation network 2001:db8::/32. The final IPv6 address for the legacy device is "2001:db8::DE00:A2:58A5".

The address is presented in the Figure 2.

```
+--------+-------+------+--------+--------+----------+---------+
| Tech.  |  U/L  | Tech. | Reserved| Mapping|  EUI-64  | Mapping |
| ID MSB |  "0"  | ID LSB|         |  MSB   | "0x0000" |   LSBs  |
|(6 bits)|(1 bit)|(5 bit)| (4 bits)|(8 bits)| (16 bits)|(24 bits)|
|  0x00  |   0   |  0x00 |   0x00  |  0xDE  |  0x0000  | 0xA258A5|
+--------+-------+------+--------+--------+----------+---------+
```

Figure 2: EIB/KNX example: the IPv6 interface identifier.

6.2. Example 2 - RFID

We assume the legacy protocol is RFID. Each RFID device is identified by its Electronic Product Code (EPC), whose length may vary from 96 to 256 bits. Let us assume to have an RFID device whose EPC is given by 01.23F3D00.8666A3.000000A05 (12 bytes). Let us assume that the RFID technology ID is "1".

We apply a hash (CRC-32) to the sequence 0x0123F3D008666A3000000A05. The result is 0xA93AFFA0.

Thereby, the IPv6 interface identifier is "0004:A900:003A:FFA0", considering the documentation network 2001:db8::/32. The final IPv6 address for the RFID tag is "2001:db8::400:A900:3A:FFA0".

The address is presented in the Figure 2.

Tech.	U/L	Tech.	Reserved	Mapping	EUI-64	Mapping
ID MSB	"0"	ID		MSB	"0x0000"	LSBs
(6 bits)	(1 bit)	(5 bit)	(4 bits)	(8 bits)	(16 bits)	(24 bits)
0x00	0	0x04	0x00	0xA9	0x0000	0x3AFFA0

Figure 3: RFID example: the IPv6 interface identifier.

7. IANA Considerations

Not yet defined.

8. Security considerations

The proposed mapping mechanism, being based on mapping proprietary protocol ID, results in such ID being incorporated in the final IPv6 address, exposing this piece of information to the Internet. The concern has been that a user might not want to expose the details of the system to outsiders. For such concern, which holds also for MAC address mapping into EUI64 addresses, please refer to appendix B in [RFC4942].

9. Acknowledgements

The authors wish to acknowledge the following for their review and constructive criticism of this proposal: Robert Cragie. Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445), and the colleagues who have collaborated in this work. In particular, Antonio Skarmeta from the University of Murcia, Peter

Kirstein and Socrates Varakliotis from the University Colleague London, and Sebastien Ziegler from Mandat International.

10. Normative References

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

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[SENSORS] Jara, A., Moreno-Sanchez, P., Skarmeta, A., Varakliotis, S., and P. Kirstein,, "IPv6 Addressing Proxy: Mapping Native Addressing from Legacy Technologies and Devices to the Internet of Things (IPv6)", Sensors 13, no. 5, 6687-6712, 2013, 2013.

Authors' Addresses

Gianluca Rizzo, Ed.
HES-SO Valais
Technopole 3
Sierre, Valais 3960
Switzerland

Phone: +41-76-6151758
Email: gianluca.rizzo@hevs.ch

Antonio J. Jara, Ed.
HES-SO Valais
Technopole 3
Sierre, Valais 3960
Switzerland

Email: jara@ieee.org

Alex C. Olivieri
HES-SO Valais
Technopole 3
Sierre, Valais 3960
Switzerland

Email: Alex.Olivieri@hevs.ch

Yann Bocchi
HES-SO Valais
Technopole 3
Sierre, Valais 3960
Switzerland

Email: yann.bocchi@hevs.ch

Maria Rita Palattella
University of Luxembourg
4, rue Alphonse Weicker
Interdisciplinary Centre for Security, Reliability and Trust
Luxembourg

Phone: (+352) 46 66 44 5841
Email: maria-rita.palattella@uni.lu

Latif Ladid
University of Luxembourg / IPv6 Forum
4, rue Alphonse Weicker
Interdisciplinary Centre for Security, Reliability and Trust
Luxembourg

Phone: (+352) 46 66 44 5720
Email: latif@ladid.lu