Light-weight multicast DNS and DNS-SD (lmDNS-SD): IPv6-based resource and service discovery for the Web of Things

Antonio J. Jara, Pedro Martinez-Julia and Antonio Skarmeta
Department of Information and Communication
Computer Sciences Faculty, University of Murcia
Regional Campus of International Excellence "Campus Mare Nostrum"
Murcia, Spain
{jara, pedromj, skarmeta}@um.es

Abstract— **Internet of Things (IoT) is presenting an enormous growing, in numbers, it is estimated that over 50 billion of devices will be connected to Internet by 2020. Therefore, it presents a high scalability requirement to manage every resource connected to the network. Therefore, It is required a high capability for autonomous registration and discovery of resources and services. In addition, it should be dynamically adapted with the inclusion of new devices in the network and changes of the existing ones. Nowadays, the most extended discovery architecture for the Internet is the Domain Name Systems (DNS), which is offering through the extensions multicast DNS (mDNS) and DNS Service Directory (DNS-SD) the query and discovery of services by type and properties. It has been already carried out some initial works on mDNS and DNS-SD for the discovery of things. Thereby, it can satisfy the discovery of resources from the IoT point of view, and discovery of services, i.e. WebServices such as CoAP from the Web of Things point of view. But, it has not been yet analyzed the impact of DNS for Smart Objects, since it cannot be directly applied, because these protocols are designed for host-based requirements, where they are not taking into account the design issues and constraints from the Smart Objects. For that reason, this paper analyzes the requirements and design issues to apply these discovery techniques in Smart Objects, carries out an overview of the satisfaction of them in the initial solutions for IoT, in order to finally offer an evaluation of different ways to apply mDNS and DNS-SD for Smart Objects, concluding with a set of recommendations and lessons learned to build a lightweight implementation of mDNS and DNS-SD for resource discovery and directory.**

Keywords- Internet of Things; M2M; Resource Discovery; DNS Service Discovery; lightweight multicast DNS.

I. INTRODUCTION

Flexibility, ubiquity, and scalability are the three required features within the current technological Era, focused on the ubiquitous computing and the Internet of Things communications [1].

Flexibility is required due to the wide range of heterogeneous environments located around the world, where the solutions required can range from simple sensors under the road for parking areas, or in the floor to measure humidity, to more complex systems, such as smart meters to measure pollution, air quality, environmental factors, and even clinical sensors for healthcare.

Flexibility, ubiquity and scalability are properties found in the current Internet, and that is why the aforementioned challenges can be solved not only with the new capabilities to link Internet with everyday sensors and devices, but also with the exploitation of data captured from the Future Internet through the so-called Internet of Things (IoT).

Future Internet and the IoT present an unprecedented growth in the number of devices and users connected to the Internet. Therefore, the devices should be as autonomous as possible in satisfying the so-called self-* functionalities, such as self-management, self-healing, and self-discovery. These properties are especially challenging in the Internet of Things, where many devices are mobile and, consequently, can change their location in the network.

For that reason, this work is focused on operating on top of the Future Internet infrastructure, i.e. IPv6. Thereby, users and clients discover and use homogenous IPv6-based resources, with protocols and technologies that are very well-known and are already deployed. Our principle goal is to avoid the out-of-IPv6-network mechanisms in order to homogenize the discovery and use of resources through the Future Internet infrastructure, i.e. through the IPv6 network for Smart Objects with technologies such as 6LoWPAN [2] and GLoWBAL IPv6 [3]. This makes services reachable through homogenous and interoperable IP-enabled technologies [4]. For example, it can be found RESTFul architectures adapted for constrained networks in protocols such as CoAP, which are allowing to reach an homogenous access through WebServices reaching the so-called Web of Things [5].

This paper is mainly focused on the discovery of the services offered by a smart object. This is very important in machine-to-machine (M2M) applications where there are no humans in the loop and static interfaces result in fragility.

This discovery of services can be conducted through network-based Information Systems that are already deployed, such as the Domain Name System (DNS) with the Service Discovery extension (DNS-SD) and multicast DNS (mDNS), or it can be defined new Resource Directories (RD) with the considerations of the smart objects requirements.

The contribution of this paper is the analysis of the capabilities to use the directory systems based on DNS, and other Discovery Servers, in order to evaluate the mechanism and possibilities to make it more lightweight and extend the current Resource Directory from M2M platforms to a more global and scalable solution.

978-0-7695-4684-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IMIS.2012.200

II. Related Works

This section is organized following the evolution of the initial requirements to connect Smart Objects, continued with the requirements to build applications over them, and finally the definition of techniques to discover services, resources and the definition of different kind of directories.

Initial works have been carried out in order to offer IPv6 connectivity to smart objects based on IEEE 802.15.4, Bluetooth Low Energy, BACNET, etc… These works are contextualized mainly under the 6LoWPAN [2], GLoWBAL IPv6 [3], and 6man [6, 7] works.

Once, it has been reached the capability to connect end-to-end through Internet to any smart object. It was considered the necessity to define an homogenous access to the application layer. Analyzing the current Internet status, Web is the most extended services medium. For that reason, it is defined the Web of Things [5], where at the beginning it was carried out RESTFul packets (HTTP) over 6LoWPAN. Since, this was seen the high flexibility and potential of these solutions. Then, it was defined a more reduced version of RESTFul for constrained environments, building the Constrained Application Protocol (CoAP).

The current status is that once we have already access to the sensors, and a set of services which can be easily and globally accessed. It is required an easy and scalable way to discover these devices and their services.

Two different levels of discovery are found following an Internet of Things and Ubiquitous computing approach for the Discovery Systems and mechanisms [8]. First, resource discovery, i.e. the discovery of devices on the network; and second service discovery, the discovery of the services, methods and functions offered by a specific resource. Usually, in Internet it is considered Resource from a general point of view, where services are part of these resources. But when the Internet is not limited to files, applications, and services, and is moved towards a more physical approach. It is required also the physical location and identification of the information.

Resources are reachable through technologies such as 6LoWPAN, GLoWBAL IPv6, Bluetooth Low Energy, or any technology offering IPv6 support.

Resource discovery is the process by which the user is able to find devices offering services according to his criteria and interests. It can differ from the resources that the user can explicitly request or from a more sophisticated discovery where the network is more pro-active, and it notifies the user about the availability of these new devices.

Resource discovery will provide descriptive information, such as the resource type or family, and some attributes to describe it. In addition, it will provide the information that the user will need to reach them, i.e.: a locator such as a URL, UID, Host Identity (HIP) or IP address.

Resource discovery management requires dynamic updates to the system with the new resources included in the network, as well as the ability to integrate the updates over mobile [9] in order to be consistent with the real resources reachable at a specific moment.

Service discovery is focused on the description of those services provided by technologies, such as those that are Web-based, i.e.: XML, Web Services, or other technologies such as JSON, and DNS Service Directory.

These services include printing and file transfer, music sharing, servers for pictures, documents and other file sharing, as well as services provided by other resources. With the expansion towards the IoT, other, simpler services can be considered, such as the environmental status consultation for temperature, humidity and lighting, a pressure value for a parking sensor, or the value of glucose for a medical device.

Different techniques can be found for resource and services discovery and the Internet of Things. Right now, the most common approach is the definition of M2M platforms, such as ThingWorx, Pachube, Sen.Se, and SENSEI, where the devices are registered in the platform, and are reachable from the Internet through WebServices such as SOAP and REST. The problem with this static approach is that it is limited to the information on the platform and the manually registered devices and, for that reason, defining more scalable solutions must be required. This makes it possible for resources entering the network to be available by registering with the discovery system, without any interaction with the user, on a directory system that is homogeneous and queriable simply over the Internet, without the need to use a specific M2M platform.

This capability for autonomous registration and the discovery functionality to be dynamically adapted with the inclusion of new devices in the network is necessary for the above-mentioned IoT to be flexible and ubiquitous. It is also necessary for the scalability required to manage every resource connected to the network, whose number is continuously increasing. It is not feasible to continue considering IoT solutions that require a manual and static management of resources, with fixed registration over specific directory systems from the M2M platforms.

Some naming systems such as Lightweight Directory Access Protocol (LDAP), Universal Description, Discovery, and Integration (UDDI), and Domain Name System (DNS) [10] offer resource and service directory capabilities, and more specific resource discovery technologies could be added, such as UPnP, JINI, Service Location Protocol (SLP), and Rendezvous or Bonjour protocol over DNS with the DNS-SD extension and multicast DNS (mDNS) [11].

However, none of the existing implementations are taking into account the requirements from the IoT in aspects such as the sleep mode, the constraints of computing power, battery capacity, available memory, or communications bandwidth that can be provided.

The next section presents the design issues for IoT, and then an analysis of the satisfaction of them from the current initial solutions for IoT. Then, we will define some optimizations and recommendations to make lightweight the mDNS and DNS-SD in order to reach a more suitable solution for the constraints and requirements from the IoT.

III. Design Issues, Requirements and Challenges

The quick evolution of the IoT is defining new challenges in term of scalability, allocation of resources and efficient discovery. Therefore it is required to define an efficient method which can be simple but yet sufficient, and satisfy the basic smart objects requirements about low-cost, lightweight, and efficiency. Therefore, it needs to be determined among the extended set of existing solutions for the resource and service discovery, which is the proper solution for the requirements and constrains from the smart objects. For that reason, this section presents the major challenges and design issues:

- **Scalable.** It is estimated that through the IoT over 50 billion of devices will be connected to Internet by 2020 [12]. It means a high number of resources, services, and locators (IPv6 addresses), which will require to be managed.
  - Therefore, it is required a decentralized architecture, such as the defined nowadays in the DNS, which allows to distribute the information about the services and location of the deployed smart objects based on their domain or anchor point. Thereby, it can be managed locally their information but accessible globally through the Internet architecture.
- **Dynamic.** The smart objects are continuously being deployed; therefore continuously new devices and services will be defined. In addition, some smart objects will be mobile (wearable systems, Intelligent Transport Systems etc.).
  - Therefore, it is required a solution, which can be easily and dynamically updated, in order to manage the creation, update and delete of entries about smart objects services and location.
- **Sleep mode.** The first constrain is that the smart objects are usually battery powered. Thus, it requires sleep, in order to optimize their battery and reach suitable lifetimes.
  - Therefore, it will limit and define new challenges for the solutions based on approaches where the endpoint is directly queried, such as the multicast DNS solution.
- **Payload size.** The original frame size from technologies such as IEEE 802.15.4 is equal to 127 bytes. 6LoWPAN has an overload of 26-41 bytes, meaning that the final available payload is reduced to half of the original size, i.e. 61 to 76 bytes from the original 127 bytes. It is reduced to less than 50% of the original frame size.
  - Therefore, it will require a high filtering of the *answers*, in order to not overload and flood to the nodes with big answers which require multiple packets (fragmentation).
- **Global query.** It needs to be defined a resource directory, which can be queried in a local level (specific domain), but also in a global level, in order to carry out wide surveys.
  - Therefore, it will require a mechanism able to offer queries in a domain-specific level, but this also should be extended with another mechanism in order to discover the domains, where are available the type of resources or devices queried.
- **Multi device operations.** It needs to be supported a mechanism to discover multiple devices as only one, for operations which are involving to multiple devices. For example, *"turn all the lights from this room on"*. Following, the payload size constrains mentioned, it is not feasible to get an answer for each bulb from a room, and after send an operation for each one of the bulbs. Therefore, it needs to be exploited multicast, and consequently, it should be also defined a support for multicast in the directory.
- **Based on existing technologies:** the access to the directory should be based on already existing mechanisms but with some considerations in order to reach a trade-off among all the presented challenges.
  - Therefore, it could be based on DNS or some of its derivations, or it could be built over the application level, i.e. CoAP.
- **Semantic description**. This needs to be defined a common description of the services, and attributes in order to carry out the queries. It is a collateral requirement to define the mechanisms to filter adequately the type of resources and services to be queried.

IV. Directory Server for Smart Objects

For smart objects, it can be found some approaches based on mDNS and DNS-SD, on the one hand, following the DNS protocol, and on the other hand defining an interface based on CoAP. These directories have in common that are focused on the location of resources and description of their services.

1) DNS Service Directory and multicast DNS

The most extended directory server in the current Internet is DNS, and an extension of the basic DNS was proposed by the IETF ZeroConf WG. Specifically, it is the DNS-SD [10] or Rendezvous protocol, which is commonly used in conjunction with multicast DNS (mDNS) in solutions such as Bonjour for MAC OS, and AVAHI for Linux OS.

DNS-SD and mDNS present a solution where no additional infrastructure, in addition to the current DNS servers, is needed, and merely requires that resources be enabled with an IP-based addressing.

The solution is focused on the re-use and extension of existing Internet standards. This can be found with a multicast approach, other protocols such as the first stage Service Location Protocol (SLP) and JINI protocols through multicast, but the advantage of mDNS is the re-use and extension of the existing Internet protocols.

In our approach, following the objective from enable all resources with IPv6 addresses, and the re-use and extension of current Internet technologies, we will focus on DNS-SD and mDNS for our proposal.

mDNS is mainly the protocol to query and populate the DNS-SD servers. This is widely used to provide Zero configuration host names (in .local domain).

This offers a distributed service discovery, since it can allocated pointers inside (mDNS) and outside (DNS-SD) of your network. DNS-SD is scalable to enterprise deployments, since it can be defined a centralized server per enterprise, building or in an IoT deployment to a room level.

The results from a DNS-SD or mDNS query are essentially identical; the same clients can work for large or small networks. Specifically, at a host-level management of the records with mDNS and with the infrastructure mode, i.e. DNS-SD, it can be also updated the repository with a modification in the delegated server, which has global consequences. The main problem for this is the cacheable DNS entries, but it can be solved defining a low lifetime for them. Thereby, it can be also dynamically adaptable.

B. Resource Directory based on CoAP (RD)

Following a similar approach to DNS-SD, it has been defined a Resource Directory (RD) based on CoAP. It is accessible through a CoAP-based interface. Thereby, it is not required to support additional protocols. This RD is used as a repository for Web Links to the resources hosted on the smart objects, which are acting as Web Servers through their REST/CoAP interfaces.

These smart objects are also able to act as clients. The RD follows a functionality very similar to the DNS, but with CoAP instead of the DNS protocol. Therefore, it is also segmented by domains and sub-domains from the naming.

These entries as a different with the common DNS, they are stateless and consequently require the refresh from the smart object (following the maximum age configuration).

In addition, this presents a mechanism similar to DNS-SD in order to carry out the query, but in this occasion it is based on the description of the parameters through the CoRE Link Format [13, 14].

Finally, the protocol [15] presents the functionalities to create, delete, and update the directory entries.

V. LIGHT-WEIGHT MDNS AND DNS-SD (LMDNS-SD)

lmDNS-SD offers a set of implementation guidelines and design recommendations in order to make suitable the use of mDNS and DNS-SD in Smart Things.

First, let us introduce the most common records from DNS:

- **A:** address record for an IPv4 address.
- **AAAA:** address record for an IPv6 address.
- **CNAME:** Alias of one name to another name.
- **NS:** For the delegation of a DNS Zone. To an authoritative name server.
- **Others:** MX for the email, HIP for the HIP identifier, LOC for the locator and others related with security stuff.

mDNS and DNS-SD are extensions of DNS with additional functionality for the records PTR and TXT. The most relevant for mDNS and DNS-SD are:

- **PTR:** Finally, this record is used for the reverse DNS lookups, i.e. from address to name. But, this presents a total different use for mDNS and DNS-SD, since it is used for the description of the services; in addition mDNS use it to filter the queries.

A usual discovery should start with mDNS protocol in local domain or DNS-SD from a more global approach, in order to discover the devices, which are offering the type of service required. For this purpose, it is used the PTR record, where we are able to define multiples pointers for a device depending on the functionality, family, type of device etc… For example, Table 1 presents multiple PTRs pointing to a light from our lab, called *light_lab*.

Table 1 PTR record for light_lab.

```
;Type
_lamp._sub._coap._udp PTR light_lab
;Services
_status._lamp._sub._coap._udp PTR light_lab
_onoff._lamp._sub._coap._udp PTR light_lab
_dimmer._lamp._sub._coap._udp PTR light_lab
;Technology
_x10._lamp._sub._coap._udp PTR light_lab
```

The discovery of these services can be carried out with a mDNS client such as AVAHI or Bonjour, in addition to the common DNS lookup services. For example, Figure 1 presents the discovery of a resource based on AVAHI, and Table 2 presents a query based on dig.

Table 2.Discovering a type of object through mDNS[1].

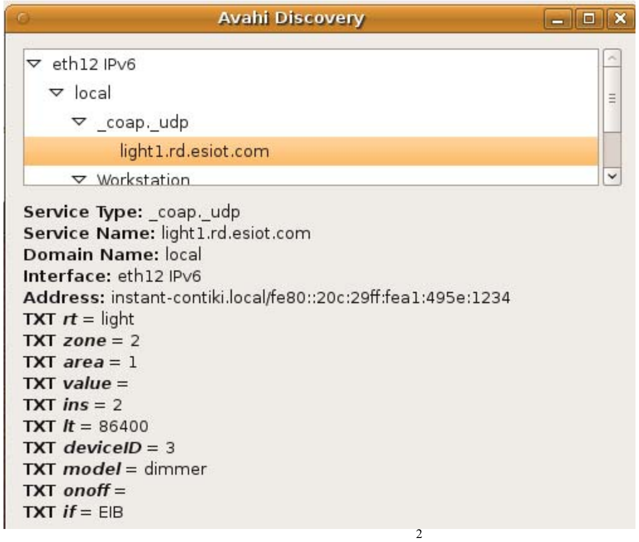
```
;_lamp._sub._coap._udp.rd.esiot.com PTR
;;  ->>HEADER<<-  opcode:  QUERY,  status:  NOERROR,
id: 62392
;;  flags:  qr  rd  ra;  QUERY:  1,  ANSWER:  1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;_lamp._sub._coap._udp.rd.esiot.com. IN      PTR

;; ANSWER SECTION:
_lamp._sub._coap._udp.rd.esiot.com.  604699  IN  PTR
light_lab.rd.esiot.com.

;; Query time: 51 msec
;; MSG SIZE  rcvd: 73
```

[1] Command details: LINUX: dig _lamp._sub._coap._udp.rd.esiot.com PTR WINDOWS: nslookup -q=ptr _lamp._sub._coap._udp.rd.esiot.com.



Service Type: _coap._udp
Service Name: light1.rd.esiot.com
Domain Name: local
Interface: eth12 IPv6
Address: instant-contiki.local/fe80::20c:29ff:fea1:495e:1234
TXT *rt* = light
TXT *zone* = 2
TXT *area* = 1
TXT *value* =
TXT *ins* = 2
TXT *lt* = 86400
TXT *deviceID* = 3
TXT *model* = dimmer
TXT *onoff* =
TXT *if* = EIB

Figure 1. Avahi discovery[2] .

Once, it is know the name of the service, which is offering what we are looking for, we required to ask for this service. Then, it is used the SRV record

- **SRV:** Generalized service location record. It is like MX but for any service. This defines which machine supports what service and on what port. The syntax is:
  SRV [priority] [capacity] [ttl] [hostname].
  Priority and capacity parameters allow to choose to the client among the different options when several hosts are offering the same service.

Table 3.Looking up de service associated to the light found.

```
;light_lab.rd.esiot.com SRV
;;  ->>HEADER<<-  opcode:  QUERY,  status:  NOERROR,
id: 6373
;;  flags:  qr  rd  ra;  QUERY:  1,  ANSWER:  1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;light_lab.rd.esiot.com.             IN      SRV

;; ANSWER SECTION:
light_lab.rd.esiot.com.       604800  IN      SRV
       0 0 1234 light1.rd.esiot.com.

;; Query time: 118 msec
;; MSG SIZE  rcvd: 79
```

This defines that the service is located at the hostname light_lab.rd.esiot.com; now we can solve the TXT entry in order to obtain more information about this device

- **TXT:** This contains metadata for the client. The format is *'[key]':'[value]'* and the contents depend on the protocol. For example, DNS-SD defines the format for these records depending on the type of records, in a similar way resource discovery from CoRE defines the link format description on these records [13].

Table 4.TXT entries with the extra information of the found light.

```
; light_lab.rd.esiot.com TXT
;;  ->>HEADER<<-  opcode:  QUERY,  status:  NOERROR,
id: 16345
;;  flags:  qr  rd  ra;  QUERY:  1,  ANSWER:  3,
AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;light_lab.rd.esiot.com.              IN      TXT

;; ANSWER SECTION:
light_lab.rd.esiot.com.       604770  IN      TXT
       "onoff\;status\;dimmer"
light_lab.rd.esiot.com.       604770  IN      TXT
       "if=X10\;housecode=A\;unitcode=5"
light_lab.rd.esiot.com.       604770  IN      TXT
       "rt=light\;ins=1\;lt=86400\;model=normal"

;; Query time: 53 msec
;; MSG SIZE  rcvd: 163
```

TXT entries are designed to be associated with the SRV entry offering extra information (metadata). Usually, it is defined a single *'[key]':'[value]'* per record such as the found in the Figure 1 in AVAHI, but in order to optimize it to reduce the number of records, it can be associated by services, resource type (rt) and interface (if) such as presented in Table 4, but if this continues presenting a high overload (176 bytes), it can be considered an unique record such as presented in Table 5, following the naming conventions that describe how services will be represented in DNS records, as defined by Web Linking description, in particular the version of Link format defined under the CoRE IETF working group [14].

Table 5.TXT query of the found light in a single TXT record.

```
; light2.rd.esiot.com TXT
;; Got answer:
;;  ->>HEADER<<-  opcode:  QUERY,  status:  NOERROR,
id: 19187
;;  flags:  qr  rd  ra;  QUERY:  1,  ANSWER:  1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;light2.rd.esiot.com.          IN      TXT

;; ANSWER SECTION:
light2.rd.esiot.com.   604800  IN      TX
       "rt=light\;ins=2\;lt=86400\;model=dimmer\;
if=EIB\;area=1\;zone=2\;deviceID=3\;value\;onoff"

;; Query time: 79 msec
;; MSG SIZE  rcvd: 130
```

Once, we have all the information about the hostname of the resource (SRV), and service description with extra information (TXT), it needs to be solved the IPv6 address of the device, which is reachable through technologies such as the aforementioned 6LoWPAN and GLoWBAL IPv6.

[2] Avahi publish command details: avahi-publish-service light1.rd.esiot.com _coap._udp 1234 rt=light ins=2 lt=86400 model=dimmer if=EIB area=1 zone=2 deviceID=3 value onoff

Table 6.AAAA query of the found light.

```
; light1.rd.esiot.com AAAA
;;  ->>HEADER<<-  opcode:  QUERY,  status:  NOERROR,
id: 60429
;;  flags:  qr  rd  ra;  QUERY:  1,  ANSWER:  1,
AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;light1.rd.esiot.com.          IN      AAAA

;; ANSWER SECTION:
light1.rd.esiot.com.   604800  IN      AAAA
        2001:720:1710::11

;; Query time: 75 msec
;; MSG SIZE  rcvd: 65
```

It can be considered A records for backward compatibility with the current Internet infrastructure based on IPv4, and also other addressing and identification spaces such as Universal Identifier (UID) from RFID or novel protocols such as Host Identity Protocol (HIP).

VI. Lesssons Learned and Recommendations to Build a Lightweight version of DNS-SD and mDNS

This section is organized following the design issues presented in the Section 3.

- **Scalable.** Such as aforementioned in the Section 4.A, DNS-SD and mDNS presents a scalable and decentralized architecture, which allows to define services in a local level through mDNS and in a global level through the hierarchical delegation of domains servers to locally managed repositories with DNS-SD. These local repositories can be located at the Border Routers from solutions such as 6LoWPAN, and consequently managed repositories even at room level.
- **Dynamic.** mDNS allows the change of the description of the services in a host-level, but even with the DNS-SD solutions, since the repositories will be allocated in a local level too, although they are visible and accessible globally allow to update the records easily and dynamically. The problem found with the dynamic changes are the DNS caches, but it can be solved defining low lifetimes, i.e. fine-grained lifetime management with the max-age attribute for the records which are susceptible of suffer changes, for example, the records associated to mobile nodes.
- **Sleep mode.** This presents high challenges for the solutions based on approaches where the endpoint is directly queried, such as the mDNS solution, especially in cases where the duty cycle is very low. For these cases, it is delegated the buffering of the requests to the coordinator and border router, in order to send it to the sensor, when this awake. Following this idea of buffering, and with the purpose of avoid that finally it needs to be required the exchange of messages with the end-node, it can be applied the DNS-SD in the Border Router, in order to offer the entries directly, instead of require to query to the end node.
  In addition, it is being considered the extensions of the directory with mirror proxy functionality also for values [16]. Thereby, the sleep nodes can delegate resource hosting to proxy, in order to make the resources available, while they are sleeping.
- **Payload size.** The original frame size from technologies such as IEEE 802.15.4 is equal to 127 bytes, and this is reduced to 61 to 76 bytes from the original 127 bytes.

  - DNS protocol usually includes a section of ***additional*** records and another section with the ***authority*** record, what means a higher overload. For that reason, such as presented in the examples from Section V, and it needs to be avoid the inclusion of these ***additional*** and ***authority*** records[3] . For example, Table 7 presents an equivalent query to the carried out in the Table 3 for the discovery of SRV entry associated to the service to consult. It can be seen that the addition of the extra information, presents a packet size of 188 bytes instead of the original 79 bytes, what means that a packet that with lmDNS-SD fills in a single frame, with the normal use of DNS-SD and mDNS requires 3 frames.

Table 7.SRV query of the found light without optimizations.

```
;; search(light_lab.rd.esiot.com, SRV, IN)
;; query(light_lab.rd.esiot.com, SRV, IN)

;; send_udp(94.142.247.17:53): sending 40 bytes
;; timeout set to 5 seconds
;; answer from 94.142.247.17:53: 188 bytes
;; HEADER SECTION
;; id = 9950
;; qr = 1  opcode = QUERY  aa = 0 tc = 0    rd = 1
;; ra = 1    rcode  = NOERROR
;; qdcount = 1 ancount = 1 nscount = 1 arcount = 4

;; QUESTION SECTION (1 record)
;light_lab.rd.esiot.com.       IN      SRV
;; ANSWER SECTION (1 record)
light_lab.rd.esiot.com.        602400  IN      SRV
        0 0 1234 light1.rd.esiot.com.

;; AUTHORITY SECTION (1 record)
rd.esiot.com.          602400  IN      NS
        rd.esiot.com.

;; ADDITIONAL SECTION (4 records)
light1.rd.esiot.com.   602512  IN      A
        155.54.210.163
light1.rd.esiot.com.   602400  IN      AAAA
        2001:720:1710::11
rd.esiot.com.          602400  IN      A
        155.54.210.159
rd.esiot.com.          602400  IN      AAAA
        2001:720:1710:0:216:3eff:fe00:9
```

[3] It can be removed with dig using the options +noauthority +noadditional

- The description of the services (TXT) should be simplified as much as possible in order to fill in a single frame. In addition, it should be defined in a unique entry following some format such as the aforementioned link format. For example, Table 4 presents the same content that in the Table 5, but simplified with the link format in a single entry.

It can be seen the difference between the usual query in Table 8 and the version of lmDNS-SD in Table 5, which is of 130 bytes instead of 221 bytes. In addition, should be more simplified the TXT entry for reduce it to values under 80 bytes, making it feasible for a single 6LoWPAN packet. This reduction could come through the use of wildcards for the identification of the parameter types, or through compression techniques such as LZ77.

Table 8.TXT query of the found light.

```
;; search(light_lab.rd.esiot.com, TXT, IN)
;; query(light_lab.rd.esiot.com, TXT, IN)

;; send_udp(94.142.247.17:53): sending 40 bytes
;; timeout set to 5 seconds
;; answer from 94.142.247.17:53: 221 bytes
;; HEADER SECTION
;; id = 24910
;; qr = 1     opcode = QUERY     aa = 0     tc = 0
rd = 1
;; ra = 1    rcode  = NOERROR
;; qdcount = 1  ancount = 3  nscount = 1  arcount
= 2

;; QUESTION SECTION (1 record)
;light_lab.rd.esiot.com.      IN      TXT

;; ANSWER SECTION (3 records)
light_lab.rd.esiot.com.       604800  IN      TXT
       "if=X10;housecode=A;unitcode=5"
light_lab.rd.esiot.com.       604800  IN      TXT
       "rt=light;ins=1;lt=86400;model=normal"
light_lab.rd.esiot.com.       604800  IN      TXT
       "onoff;status;dimmer"
;; AUTHORITY SECTION (1 record)
rd.esiot.com.         604800  IN      NS
       rd.esiot.com.
;; ADDITIONAL SECTION (2 records)
   rd.esiot.com.              604800  IN      A
       155.54.210.159
rd.esiot.com.         604800  IN      AAAA
       2001:720:1710:0:216:3eff:fe00:9
```

- **Global query.** DNS-SD is already accessible globally, but it continues requiring the specification of the domain under which to carry out the query, in order to make it more scalable and be able to discover the domain, where are available resources of the type that we are interested. Our current ongoing work is focused on the definition of a P2P architecture based on a overlay built with chord over the lmDNS-SD architecture, in order to discover the DNS-SD directories and domains of interest through the Distributed Hash Tables (DHT) from the different domains [17].
- **Multi device operations.** In order to support multiple devices, it can be defined in the DNS-SD additional entries based on multicast. Thereby, the query for *_alllights* will point to a multicast address which will be linked to all the lights from that room, building or domain. Some examples of multicast for building control are found in [18].
- **Based on existing technologies:** It is based on DNS, and its extensions DNS-SD and mDNS.
- **Semantic description**. This needs to be defined a common description of the services, and attributes in order to carry out the queries.

For this purpose, it is being defined from the IPSO Alliance a common family of interfaces and resource types for the resource directory from CoRE [19]. Therefore, it could be re-used in a similar way as it is re-used the link format.

As an alternative, it can be defined more complex solutions such as Triple Spaces on RDF [20], which also allows to retrieve, create, modify or delete resources in the RDF graphs, and this representation of the knowledge is based on a common ontology, which shares all the entities involved in the communication. The queries over RDF can follow a pattern similar to the defined in CoAP based on triple pattern with wildcards (e.g. ?s) or also more sophisticated and complex solutions such as SPARQL. It can be also applied with the description of devices through Device Profile for Web Services (DPWS), based on SOAP or REST for the Web of Things [21].

Finally, our on-going work is mainly focused on the research about the capabilities to integrate the description of resources through RDF in the TXT records from DNS, and the query through mDNS following the described PTR patters as if it was wildcards. For the description of the resources, it will be followed GSN and SPITFIRE works [22].

In summary, regarding the semantic description, we are mainly interested on offering a semantic layer for sensor discovery and provisioning and how we can integrate CoAP and 6LoWPAN solution in this framework through the overlay discovery based on chord or resource directory based on DNS-SD. In that sense that could certainly fit with OpenIoT and SPITFIRE approaches.

VII. Conclusions and Future Work

The initial works under the Internet of Things (IoT) and Web of Things (WoT) were focused mainly on establishing connectivity in a variety of challenging and constrained networking environments through techniques such as 6LoWPAN, GLoWBAL IPv6, and lwIP. After this, it was focused on offer access for the definition of applications through WebServices following the REST style with CoAP. Once, it has been already offered the capabilities to access globally to a resource and interact with this through IPv6 and CoAP. It needs to be offered a mechanism to discover resources and services.

The most extended protocol for the discovery of resources and services in the current Internet is DNS and its extensions mDNS and DNS-SD. Therefore, at the same way that IPv6 has been adapted for constrained environments with 6LoWPAN and GLoWBAL IPv6, RESTFul with CoAP, XML with EXI, among others. It has been started to consider the re-use and extension of DNS for the resource directory of the IoT and Web of Things. The most relevant work is try to offer an interface for resource directory based on CoAP instead of DNS, but in this work we consider relevant to continue using the same Internet mechanisms. Therefore, it has been defined how to use the current DNS, DNS-SD and mDNS in a light-weight and optimal way considering the constraints and requirements from the Smart Objects.

It has been presented lmDNS-SD, which is a set of lessons learned and recommendations in order to fulfill the design issues and requirements from Smart Objects. This is mainly focused on, first, reduce the number of records from *additional* and *authority* sections, second, reduce the number of TXT records with the definition of an unique record based on link format from CoRE, third, the compression of the TXT entries with wildcards and other compression mechanisms such as LZ77, four, the adjustment of the max-age from the entries in order to make the changes dynamic for DNS caches, and other pending issues such as offer a global discovery of the domains which contains resources of a specific type through an overlay built with Chord, which is our ongoing work, and finally, also the introduction of RDF in TXT entries in order to include a more sophisticated semantic description of the things, which could be exploited by some servers through complex query solutions such as SPARQL.

ACKNOWLEDGMENT

This work has been made possible by the means of the Excellence Researching Group Program (04552/GERM/06) from Foundation Seneca, FPU program (AP2009-3981) from Education and Science Spanish Ministry, with funds from the frames of the IoT6 European project (STREP) from the 7th Framework Program (Grant 288445).

REFERENCES

[1] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey". Computer Networks Vol. 54, No. 15, pp. 2787-2805, 2010.

[2] J. Hui, and P. Thubert. “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Network”. IETF 6LoWPAN Working Group, RFC6282, 2011.

[3] A, J. Jara, M. A. Zamora, and A. Skarmeta, “GLoWBAL IP: an adaptive and transparent IPv6 integration in the Internet of Things”, Mobile Information Systems, “in press”, 2012.

[4] Joel J. P. C. Rodrigues and Paulo A. C. S. Neves, “A Survey on IP-based Wireless Sensor Networks Solutions”, in International Journal of Communication Systems, Wiley, ISSN: 1074-5351, Vol. 23, No. 8, pp. 963-981, August 2010.

[5] Z. Shelby, "Embedded web services," Wireless Communications, IEEE, Vol.17, No.6, pp. 52-57, doi: 10.1109/MWC.2010.5675778, December 2010

[6] Kerry Lynn, Jerry Martocci, Carl Neilson, Stuart Donaldson. “IPv6 over MS/TP Networks”, draft-ietf-6man-6lobac-01,6man working group, 2012.

[7] Charles Frankston, BACNET discovery based on DNS, BACnet IT strawman proposal, 2009.

[8] W.K. Edwards, "Discovery systems in ubiquitous computing", Pervasive Computing, IEEE, Vol.5, No. 2, pp. 70- 77, doi: 10.1109/MPRV.2006.28, 2006.

[9] S. Kiyomoto, and K. M. Martin. “Model for a Common Notion of Privacy Leakage on Public Database”. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), Vol. 2, No. 1, pp. 50-62, 2011.

[10] S. Cheshire, and M. Krochmal. “DNS-Based Service Discovery”, IETF Zeroconf Working Group, www.zeroconf.org/ and www.dns-sd.org/, draft-cheshire-dnsext-dns-sd.txt, 2011.

[11] Cheshire, S. and M. Krochmal, "Multicast DNS", draft-cheshire-dnsext-multicastdns-15 (work in progress), December 2011.

[12] B. Emerson, “M2M: the internet of 50 billion devices”, Win-Win, Editorial: Huawei, January 2010.

[13] Z. Shelby, "CoRE Link Format", draft-ietf-core-link-format-06, IETF work in progress, June 2011.

[14] Lynn, K. and Z. Shelby, "CoRE Link-Format to DNS-Based Service Discovery Mapping", draft-lynn-core-discovery-mapping-01 (work in progress), July 2011.

[15] Krco, S. and Z. Shelby, "CoRE Resource Directory", draft-shelby-core-resource-directory-02 (work in progress), October 2011.

[16] Vial, M., "CoRE Mirror Proxy", draft-vial-core-mirror-proxy-00 (work in progress), March 2012.

[17] Stoica, Ion, Morris, Robert, Karger, David, Kaashoek, M. Frans and Balakrishnan, Hari, “Chord: A scalable peer-to-peer lookup service for internet applications”, Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, New York, NY, USA, pp. 149--160, 2001.

[18] P. van der Stok, K. Lynn, A. Brandt. “CoRE Discovery, Naming, and Addressing”, draft-vanderstok-core-dna-01, (work in progress), March 2012.

[19] Shelby, Z. and M. Vial, "CoRE Interfaces", draft-shelby-core-interfaces-02 (work in progress), March 2012.

[20] Gómez-Goiri, A., Emaldi, M., López-de Ipiñaa, D. “A semantic resource oriented middleware for pervasive environments,” UPGRADE journal, vol. 2011, Issue No. 1, pp. 5–16, feb 2011.

[21] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services", Services Computing, IEEE Transactions on, Vol.3, No.3, pp.223-235, doi: 10.1109/TSC.2010.3, 2010.

[22] Dennis Pfisterer, Kay Römer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, Henning Hasemann, Alexander Kröller, Max Pagel, Manfred Hauswirth, Marcel Karnstedt, Myriam Leggieri, Alexandre Passant, and Ray Richardson. “SPITFIRE: Toward a Semantic Web of Things”, IEEE Communications Magazine, pp. 40-48, November, 2011.