CoRE

Internet Draft
Intended status: Experimental
Expires: September 2019

B. Djamaa
A. Yachir
EMP University

March 31, 2019

Proactive Discovery of CoRE Resource Directories
draft-djamaa-core-proactive-rd-discovery-00.txt

Abstract

The CoRE working group has proposed a Resource Directory (RD) solution to facilitate the discovery of the resources provided by constrained sensor and actuator networks. For such a mechanism to be effectively deployable, endpoints must first discover the existence of an RD in the network before being able to exploit its functionalities. This document presents Proactive RD Discovery (PRD); a scalable and effective mechanism to discover RDs. To achieve such qualities, PRD follows an announce-based model that builds upon CoAP Group Communication. PRD aims to provide important performance in terms of energy consumption, generated traffic, expressivity, and RD discovery time.

This Internet-Draft will expire on September 31, 2019.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Context 3
1.2. Terminology 4
1.3. Motivations 4
2. Background on Group Communication..... 6
3. PRD Overview 6
4. PRD Operations 8
4.1. Directory advertisement messages 8
4.1.1. DA generation and forwarding 8
4.1.2. DA Processing 10
4.1.3. Registration Maintenance and Update 11
4.2. Directory Solicitation messages 12
4.2.1. DS Generation and Forwarding 12
4.2.2. DS Processing 13
4.3. Directory Registration Removal 13
4.3.1. DR Generation and Forwarding 13
4.3.2. DR Processing 14
5. Multiple RD Discovery 14
6. Security Considerations 15
7. IANA Considerations 15
8. References 15
8.1. Normative References 15
8.2. Informative References 15
9. Acknowledgments 16

1. Introduction

1.1. Context

The Constrained Application Protocol (CoAP) [RFC7252] provides a unified mechanism to exchange application data in LLNs. Thus, extending today's Web towards a Web of things. CoAP builds upon the Representational State Transfer (REST) design paradigm to achieve its objectives. Indeed, in CoAP-enabled IoT applications, each sensor/actuator node is basically seen as an endpoint, exposing sensor readings, actuating capabilities and internal information as REST resources that can be queried by clients acting on behalf of user applications.

In order to discover the REST resources provided in LLNs, seamless and automatic discovery of available resources is an imperative. Such resource discovery solutions can be achieved using a multitude of techniques depending on a number of parameters including network size, application requirements and available infrastructure. For instance, fully distributed solutions [RFC6690],[RFC6762], and [RFC6763] can be well suited for an infrastructure-less, small-size, zero-configurable IoT network, while a centralized solution [CORE-RD] might be deployed for large-scale IoT networks, having dedicated resource-rich discovery servers.

In this context, CoRE has proposed a resource directory solution[CORE-RD] responding to SD requirements in large-scale, infrastructure-based RESTful IoT applications. [CORE-RD] defines a Resource Directory (RD) where resource providers register their available resources for clients to query. The RD follows the generic architecture of centralized discovery mechanisms. In such architecture, providers register the description of their public resources at the directory by issuing POST requests. The directory confirms the registration for the specified period and returns its location to the provider. Clients then query the RD by issuing GET requests looking for descriptions matching their requests. The default description format adopted by the RD is the CoRE link format [RFC6690], which is carried as a payload in a CoAP message. Such description has many resource attributes including resource type (rt), interface description (if) and path. To achieve RD operations, new attributes have been defined in [CORE-RD] such as the endpoint attribute (ep) specifying the endpoint hosting the resource registered in the RD, and the lifetime attribute (lt) indicating a valid registration period. A base URI attribute (base) is also introduced in order to allow an endpoint to specify the scheme, ip-address and the port on which it will be accessible. Such base attribute is of great importance to this document since it is reused

by the RD to advertise itself. The RD makes registered resources
 attribute) issues a GET request to the RD in the following format
/.well-known/core{?search*}, with the filter {?search*} containing
known attributes about the required resources.

As a key aspect of such architecture, both clients and providers must first discover an RD before being able to exploit its services. While the resource directory draft envisages many reactive mechanisms to achieve RD discovery, the default mechanism adopted by [CORE-RD] is the CoAP's resource discovery mechanism [RFC6690]. [CORE-RD] proposes other techniques for RD discovery, which are informed guesses, and lets it open, however, to develop other mechanisms in order to achieve RD discovery. This draft presents a first proactive mechanism for discovering RDs based on CoAP Group Communication.

1.2. Terminology

This document uses the following terms and abbreviations:

RD: Resource Directory as defined in [CORE-RD]

PRD: Proactive RD Discovery

Endpoint (EP): as defined in [RFC7252], describes a CoAP server or client.

CoAP Group: A set of CoAP endpoints subscribed to the group's associated multicast address. An endpoint May join different CoAP Groups. Group Membership is dynamic. As by Section 12.8 of [RFC7252], all CoAP Endpoints SHOULD join "All CoAP Endpoints" for both the IPv6 link-local address ff02::fd and the site-local address ff05::fd.

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 [RFC2119].

1.3. Motivations

Section 4 of the current resource directory draft [CORE-RD] envisages and describes 7 approaches for a device to achieve RD discovery depending on three mains cases:

1. The device is configured with a way to find RD:

- Approach 1 - Configure the device with a specific IP address for the RD.

- Approach 2 - Configure the device with a DNS name for the RD and use DNS to return the IP address of the RD.

- Approach 3 - Configure the device with a service discovery mechanism like DNS-DS.

2. The network provide a specific configuration:

- Approach 4 - Piggyback RD IP address during neighbor discovery phase using a new RDAO defined in Section 4.1.

- Approach 5 - Piggyback RD IP address during DHCP phase.

3. Neither the device nor the network offer any specific configuration:

- Approach 6 - 6LBR in a 6LoWPAN acts as a RD and the device can send a unicast query to 6LBR to confirm its RD function.

- Approach 7 - Use CoAP multicast query to find a RD in a network which supports multicast.

Approaches 1-3 can be applied to both local and wide area networks, while approaches 4-7 are more applicable to networks such as 6LoWPANs [RFC4944]. Besides, Approaches 1-2 necessitate hard configurations in the device making it hard to deal with network dynamics (e.g., change of RD addresses, RD port, Number of RDs, etc.). Approach 3 necessitates also hard configurations if used with unicast DNS or needs to rely on IP Multicast when used with mDNS [RFC6762], which does not work beyond link-local scope.
For the second category of solutions, Approach 4 allow an RD to announce its address using ND with no option to specify the port and/or the scheme employed by an RD. Approach 5 uses DHCP instead of ND to announce RD address, but no option is currently defined for such task.
The last category of solutions do not rely on any device reconfiguration nor any assumption on available network primitives. Thus, approach 6 assumes that 6LBR plays the role of RD. Besides the inconvenient of imposing that the RD be physically integrated with the border router, approach 6 may create many issues including single point of failure and significant load incurred by endpoints

near the border router. Finally, approach 7 relying on CoAP service discovery [RFC6690] has the advantage to respond to all RD discovery requirements without any device hard configurations or assumed network infrastructure. It is the default right now. All examples in [CORE-RD] are given using this approach. However, such mechanism may result in excessive network resource consumption [PRD].
Moreover, RD discovery can be performed in two ways: query-based on-demand); or announce-based (proactive). Only ND and DHCP-based approaches that are announce-based. Five out of the above 7 approaches proposed in [CORE-RD] are query-based. This discovery way can generate excessive network overhead along with RD bottlenecks when there is a great number of devices in the same network looking for RDs either for publication of their resources or discovery of available resources [PRD]. In addition, in all approaches, there is no available clues for endpoints to choose which RD to use (e.g. supported content-format...etc.). This implies that endpoints should check for all available RDs. Finally, there is a price in terms of RD discovery time to pay in all these approaches.
To deal with aforementioned issues, we propose Proactive RD Discovery (PRD), an announce-based RD discovery mechanism using CoAP Group Communication. PRD aim to address all issues above.

2. Background on Group Communication

TBD.

3. PRD Overview

PRD allows a resource directory to proactively advertise its presence and provided capabilities for endpoints to cache locally and use for the sake of RD finding. The proposal makes use of CoAP messages and methods to achieve proactive discovery of an RD. It particularly adopts POST requests to proactively announce (push) RD information to the network in a CoAP message called Directory Advertisement (DA) as depicted in Figure 1. The pushed RD information MUST support CoRE Link Format [RFC6690] and can be also pushed in other formats.PRD implementations using this specification MUST support the application/link-format content format (ct=40).

Such message is POSTed to the "All CoAP Endpoints" site-local scoped IP address using CoAP Group Communication [RFC7390]. At the reception of the first DA, an endpoint creates a registration containing advertised RD information, which is kept for a specified lifetime, updated using either PUT or POST methods, and propagated in a wavelike pattern from endpoints near the RD to those at the

edge of the network as shown in Figure 1. Using CoAP Group Communication over IP Multicast ensures that, with time, all endpoints will receive the RD's DA message. However, an endpoint requiring to use RD services before receiving the DA can issue a link local multicast Directory Solicitation (DS) GET request looking for resources having the attribute rt = core.rd* (Section 4.2). Upon reception of a DS request, an endpoint having matching RD information may issue a response. Finally, PRD envisages a state maintenance mechanism (Section 4.3) providing seamless reactions to network dynamics. Indeed, in PRD, all POSTed RD information is only kept for the specified lifetime and must be periodically refreshed by the RD. Besides, the RD can explicitly remove such information by issuing Directory registration removal (DR) message.

This PRD functioning allows an RD to announce itself directly to endpoints within a link-local, realm-local and/or site-local domain. If the RD happens to be separated from the endpoints via an LLN Border Router (LBR), for example, The Resource Directory Address Option (RDAO) using IPv6 Neighbor Discovery (ND) introduced in section 4.1.1 of [CORE-RD] can be used. Indeed, RDAO carries information about the RD address up to the LBR, which can then advertises it to the endpoints using PRD.

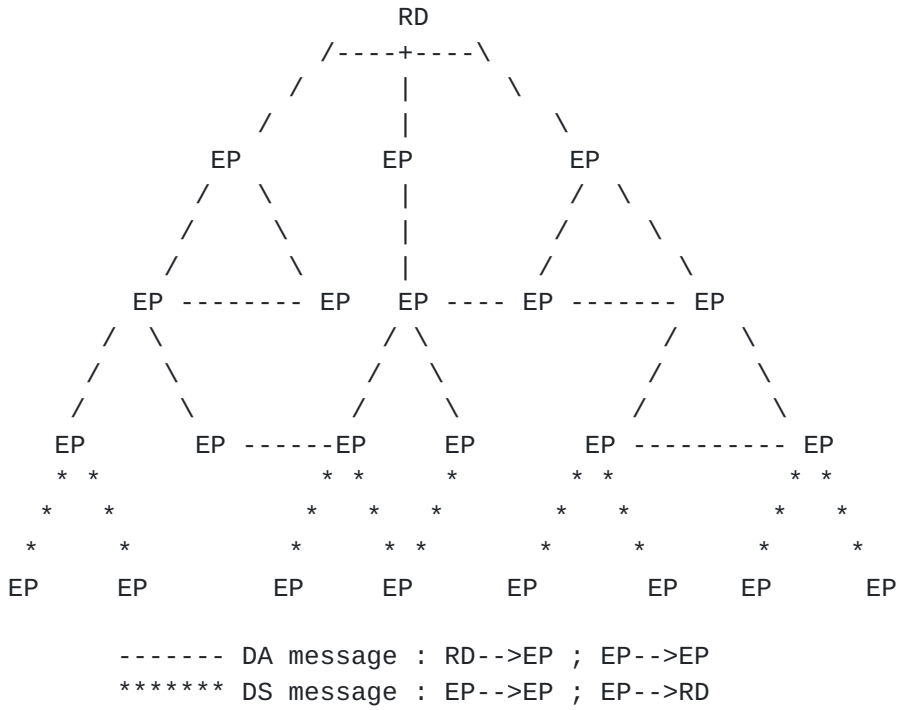

Figure 1: An Overview of proactive RD discovery

PRD support endpoints able to use the registration interface and Simple registration interfaces of [CORE-RD]. Endpoint that cannot support both interfaces and are being registered in the RD via third-party commissioning tools (section 5.2 of [CORE-RD]) may simply ignore DA messages.

Finally, PRD allows to find the RD along with its URIs at once, saving, thus, one RD discovery cycle.

4. PRD Operations

This section presents the details of PRD operations. These are categorized into three basic functionalities:

- Generation, forwarding, processing and maintenance of DA messages,
- Generation, forwarding and processing of DS messages,
- Generation, forwarding and processing of DR messages,

Each operation is detailed in the following subsections

4.1. Directory advertisement messages

DA message uses the non-idempotent CoAP method POST. To be compliant with [RFC7390], this draft has designed the resource to be posted (resource containing RD information) to cope with the unreliable nature of LLNs. Indeed, being based on Multicast protocols dedicated for LLNs such as MPL increases the reliability of multicast discovery thanks to periodic repetitions. Moreover, missing the POSTed resource by some endpoints does not compromise the operations of PRD. For instance, the Directory Solicitation message discussed below (Section 4.2) allow such node to ask for the information when needed. In a worst case scenario, the endpoint may ask the RD directly for such information using approach7 (Section 4.3) in [CORE-RD]. In the other case, where the POST request is received multiple times, an endpoint SHOULD discard subsequent registrations. Knowing that a DA message from the same RD is redundant can be done via the Message-ID field or via the endpoint attribute contained in the request.

4.1.1. DA generation and forwarding

The RD generates and maintains its DA messages using CoAP Group Communication. A new DA, with a new Message-ID, is generated by the RD when:

- The RD becomes enabled, reboots and/or when the lifetime of the previous DA is about to expire;

- The IP address and/or port number on which the RD is accessible change;

- Each time any advertised RD resource attribute and content changes (e.g., a change in the RD path).

The DA message is then POSTed to the "All CoAP Endpoints" site-local scoped multicast address [RFC7390] to be registered on endpoint(one registration is done per RD).The registration request interface is specified as follows:

Interaction: RD ->EPs
Method: POST

Content-Format: application/link-format or any other indicated media type representing web links
URI Template: {+rd}{?ep,lt,base,extra-attrs*}

URI Template Variables:

rd := RD default registration URI (mandatory). This can bea default location inferred from "ep" and/or "base", for instance, /rd1.

ep := RD name, which is indicated using the "ep" attribute introduced in [CORE-RD] and has the same specifications (mandatory). It MUST be unique and should be used to uniquely identify an RD.

lt := Lifetime (optional). This attribute indicates the lifetime of an RD registration in the range of 60-4294967295 seconds, similarly to [CORE-RD]. If absent, a default value a default value of DEFAULT-RD-REGISTRATION is assumed.

base := Base URI (optional). This parameter is defined similarly to [CORE-RD] and convoys the scheme, ip-address and port on which the RD is accessible. If absent, this information can be inferred from the scheme of the protocol, source ip-address and source port of the registration request. The Base URI can be then constructed in the same way as in [CORE-RD].

extra-attrs*: = additional attributes that MAY be defined in future specifications.

The example below shows an RD with the name "rd1" POSTing its information to the network using the above interface.

```
Req: POST coap://[All CoAP Endpoints:DPort]/rd?ep=rd1&
base=coap://rd1-address:rd1-port
Content-Format: 40
Payload:

</rd>; rt=core.rd,

</rd/res>; rt=core.rd-lookup-res,

</rd/ep>; rt=core.rd-lookup-ep
```

Responses to this DA message are suppressed.

DAs are POSTed periodically each DA-ANNOUNCE-PERIOD. This period can be tailored to the needs of the application, if not specified a default value of 25 hours can be assumed. Furthermore, this period can be adapted to network dynamics.

4.1.2. DA Processing

Receiving a DA message from an RD causes an endpoint to update its registration of RD information. To do so, the following rules apply for a POST request targeting an rd.

- When the ep value, uniquely identifying an rd, contained in the registration request is different from any existing registration value, a new registration is generated.

- When the ep value, uniquely identifying an rd, contained in the registration request matches an existing registration, the content and parameters of the existing registration are updated with the content of the registration request.

In the former, the endpoint creates a registration at an assumed default path that can be inferred from the "ep" and/or "base" attribute to ensure its uniqueness per RD. In the latter, the exiting registration SHOULD only be updated. Indeed, this interface MUST be implemented to be idempotent. In either case, the endpoint suppresses responses to DA messages following CoAP [RFC7252] and hence CoAP group communication [RFC7390] recommendations. Exceptionally, endpoints unable to use the assumed default location for RD registrations MAY respond with the location chosen. Processing of such responses by RD is TBD.

The endpoint then adds the created resource into its /.well-known/core resource for the sake of making it discoverable to other endpoints issuing DS messages.

4.1.3. Registration Maintenance and Update

Each endpoint keep one registration per RD. A registration is represented by a set of links in the CoRE link format as specified in [RFC6690]. It has the following mandatory and optional attributes:

- A mandatory RD name convoyed using the "ep" attribute introduced in [CORE-RD] and has the same specifications;

- A mandatory registration Base URI,"base", in the format scheme://authority part. "base" MAY have multiple values depending on the supported RD addresses, ports and schemes. An endpoint may use different base URIs to contact the RD depending on the supported functionalities;

- A mandatory lifetime, "lt", in seconds;

- Optional web links describing the resource.

An RD registration is kept active for the specified lifetime. It is the responsibility of an RD to refresh its registration and confirms its presence in the network by issuing DA messages periodically using the registration interface of section 4.1.1. This update, however, SHOULD only contain the content and parameters that have been changed. If nothing has been changed from the previous registration, the RD issues a DA only containing the "ep" parameter. Receiving such a DA causes the endpoint to refreshes the registration for the initial "lt" value.

An endpoint SHOULD mark an expired non-updated RD registration as stale and SHOULD NOT respond to DS messages (Section 4.2) concerning this RD. The endpoint may turn a stale registration into the active state every time it has a successful interaction with the RD. Because of this, the RD MAY adapt the value of DA-ANNOUNCE-PERIOD depending on the interactions it has with endpoints. Specifying such a behavior is TBD.

To deal with the unreliability of group communications, an endpoint not receiving DA to refresh an about to expire RD registration, can issue a DS message (Section 4.2.1) to ask for such information from its neighboring endpoints. If a successful response is received, it will update its registration with the lifetime indicated in the "lt"

parameter. Otherwise, it MAY contact the RD directly by unicast using the information from the expired registration or via multicast using approach 6 of [CORE-RD] to GET such information. The response to such a GET request will cause the node to refresh its registration. An endpoint, MAY, however, decide to delete an RD registration if it fails to reach the RD.

4.2. Directory Solicitation messages

4.2.1. DS Generation and Forwarding

In case of missing a DA, on-demand directory solicitations can be issued using the DS message. In addition to speeding up RD discovery, this functionality is particularly important for new endpoints joining a network. For instance, a new node joining the network can discover the RD by issuing a DS message as shown in Figure 1. DS messages are sent a MAX-DS-TRANSMISSIONS times separated by a DS-TRANSMISSION-INTERVAL. If no application values are devised for MAX-DS-TRANSMISSIONS and DS-TRANSMISSION-INTERVAL, default values of 3 time and 10 seconds, respectively, can be used. If still there are no responses, the originator switches to a slower transmission rate. The transmission of a DS is cancelled by receiving a response or a DA containing the requested information. The DS template is follows the rule of [RFC6690] and is given below:

Interaction: EPs->EPs and RDs
Method: GET
URI Template: /.well-known/core{?rt&base&ep}
URI Template Variables:

rt := Resource Type (mandatory). SHOULD contain one of the values "core.rd", "core.rd-lookup*", "core.rd-lookup-res", "core.rd-lookup-ep", or"core.rd*"

base := Base URI (optional). SHOULD contain a value from "base" values of existing RD registrations of the issuing endpoint.

ep := RD name (optional). SHOULD contain a value from "ep" values of existing RD registrations of the issuing endpoint.

Accept: absent, application/link-format or any other supported media type.

DS messages are sent to the "All CoAP Endpoints" link-local scoped address, no forwarding is required. The example below shows an endpoint issuing a DS message looking for "rt=core.rd".

DS: GET coap://[ff02::fd]/.well-known/core?rt=core.rd

DS Res: 2.05 Content
</rd>; rt="core.rd"; base="coap://rd-address:rd-port"

DS messages looking for specific RD name and base URIs may be issued by endpoints having such parameters searching for a specific RD to refresh its registrations for instance.

4.2.2. DS Processing

On receiving a DS message, an endpoint having matching resource registrations generates a unicast response to be sent back to the DS originator following the specification of [RFC6690].

Since multiple endpoints might respond to a multicast DS, the congestion control mechanism suggested by CoAP [RFC7252] should be used. Thus, endpoints should insert a random delay, called leisure time, before issuing their responses. The lower bound of the leisure time can be approximated based on an estimate of the group size G, the data transfer rate T and the response size S as follows: LBLeisure = S*GT. If endpoints are not able to estimate such parameters, a default value of 5s might be used.

4.3. Directory Registration Removal

Being a proactive approach, PRD might suffer from network inconsistencies by keeping information about an RD which is no longer available. The mechanisms discussed in section 4.1.3 concerning registration maintenance and updates can achieve soft deregistration of stale information. Additionally, this section, introduces an explicit Directory Registrations Removal (DR) mechanism. Indeed, an RD going to be unavailable SHOULD explicitly announce its unavailability to the network. To do so, the RD SHOULD issue DR messages to the "All CoAP Endpoints" site-local scoped IP address using CoAP Group Communication with the DELETE method.

4.3.1. DR Generation and Forwarding

DR enables a directory to advertise its graceful disappearance. This is done by issuing a DR message with the DELETE method to the path inferred from the "ep" and/or "base" attribute of this RD, which is unique per RD. This functionality is ensured using the removal interface specified as follows:

Interaction: RD -> EPs
Method: DELETE

URI Template: {+location}
URI Template Variables:

location := is the assumed default location, which MAY be inferred from the "ep" and/or "base" attribute.

Similarly to DA messages, responses to DR messages are suppressed.

The following example shows the removal of an RD registration located at the "/rd1" path.
DR: DELETE coap://ALL COAP ENDPOINTS/rd1

The forwarding of this message is done using CoAP group communication similarly to DA (Section 4.1.1).

4.3.2. DR Processing

The reception of a DR message with the DELETE method causes the endpoint to delete corresponding RD information and hence, suppresses it for its /.well-known/core resource.

5. Multiple RD Discovery

Having multiple RDs might be preferable not only for redundancy reasons but also to support and provide different services. Indeed, a provider might be willing to register with an RD that supports a specific content-format/protocol. Similarly, a client might prefer to query an RD supporting its required parameters. PRD provides support of the discovery of multiple RDs in a network. To do so, each RD announce its available interfaces and supported functionalities using CoAP Group communication separately. This requires endpoints to keep and maintain a resource per RD. This might increase the burden on sensor/actuator endpoints. However, taking into account the number of expected RDs in a network, this solution may present a fair trade-off. Optimized versions for supporting proactive announcements of multiple RDs may be devised and are out of the scope of this draft.

Finally, it should be noted that by supporting discovery of multiple RDs, PRD can play a pivotal role into distributing hints allowing both providers and clients to be aware of available resources and functionalities of each RD and giving them all necessary information to access such resources. For instance, PRD can distribute information about the content formats supported by an RD using the content type (ct) attribute. Clients and providers might use this information to select preferred content formats for interacting with RDs.

6. Security Considerations

TBD

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <https://www.rfc-editor.org/info/rfc7390>.

[CORE-RD] Shelby, Z., Koster, M., Bormann, C., Stok, P., and C. Amsuess, "CoRE Resource Directory", draft-ietf-core-resource-directory-20 (work in progress), March 2019.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

8.2. Informative References

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <http://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4

Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[PRD] Djamaa, B. and Yachir, A.: A Proactive Trickle-based Mechanism for Discovering CoRE Resource Directories. Procedia Comput. Sci. 83, 115-122 (2016).

9. Acknowledgments

The authors would like to thank Chonggang Wang and Akbar Rahman for fruitful inputs and discussions.

Authors' Addresses

Badis Djamaa
EMP University
Bordj-el-Bahri, Algiers
Algeria

Email: badis.djamaa@gmail.com

Ali Yachir
EMP University
Bordj-el-Bahri, Algiers
Algeria

Email: a_yachir@yahoo.fr