# Resource Design in Constrained Networks for Network Lifetime Increase

N. Correia, A. Mazayev, G. Schütz, J. Martins and A. Barradas

***Abstract*—As constrained "things" become increasingly integrated with the Internet and accessible for interactive communication, energy efficient ways to collect, aggregate and share data over such constrained networks are needed. In this article, we propose the use of CoRE Interfaces to build resource Collections having a network lifetime increase in mind. More specifically, based on existing atomic resources, Collections are created/designed to become available as new resources, which can be observed. Such resource design should not only match client's interests, but also increase network lifetime as much as possible. For this to happen, energy consumption should be balanced/fair among nodes so that node depletion is delayed. When compared with previous approaches, results show that energy efficiency and network lifetime can be increased while reducing control/registration messages, which are used to set up or change observations.**

***Index Terms*—Internet of Things, constrained networks, energy saving, fairness, CoRE interfaces.**

## I. Introduction

THE *Internet of Things* (IoT) provides a bridge between the physical and digital worlds, allowing the development of innovative applications and services. For this reason, IoT is receiving enormous attention among researchers, industry and governments. This new wave will bring into the Internet all kinds of "smart things" (e.g., sensors), allowing multiple users and applications to access them [1], [2]. The IoT will change the way all businesses, governments and consumers interact with the physical world, so it has been called the next Industrial Revolution.

In the IoT sensors can be organized into networks and, for this reason, better ways to collect, aggregate and share data over the network are needed. Having such goal in mind, the *Constrained RESTful Environments* (CoRE) working group, within *Internet Engineering Task Force* (IETF), focused on the development of a resource-oriented application framework so that data can be stored, retrieved and manipulated using a client-server protocol [3]. *The Constrained Application Protocol* (CoAP), a Web application transfer protocol is the main result of this group. The Observe extension to CoAP also emerged to give clients the ability to observe resource changes [4]. That is, the client keeps its resource representation updated over time. When compared to the *Hypertext Transfer Protocol* (HTTP), CoAP/Observe greatly reduces the overhead in implementation complexity, bandwidth requirements and latency, while increasing reliability as a result of the publish/subscribe model used [4], [5].

In CoRE, the possibility of discovering resources hosted by constrained servers becomes important for applications to run without human intervention [6]. For this purpose, Web discovery and linking in constrained environments has been specified, and Interface Types for resource design are now on an ongoing standardization process [7]. These Interface Types, based on CoRE standards for information representation and exchange, allow the server side to compose/organize resources and the client side to discover and determine how to consume such resources. A client discovering the interface ("if" link attribute) will be able to consume resources based on its knowledge on Interface Types.

Resource discovery allows application clients to adapt to different resource organizations, without having a previous knowledge of the hosted data structures. However, such resource organization is done having no knowledge of resources hosted by other CoAP servers, and no knowledge of observation requests throughout all the constrained network. This means that resources are made available at hosts/servers with no overall network goal in mind (e.g., energy saving). A client observing multiple resources at different locations would receive multiple independent notifications, which travel throughout the network independently. The definition of Interface Types brings, however, the possibility of dynamically adapting the resources available at hosts/servers to an overall goal, like increasing the network lifetime. More specifically, link Collections of original resources, and/or other Collections, can become available in order to reduce the overall signaling required to set up observations, and to reduce the number of notifications. This leads to energy saving, which is critical for the future availability of resources.

N. Correia, A. Mazayev, G. Schütz, J. Martins and A. Barradas are with the Center for Electronic, Optoelectronic and Telecommunications (CEOT).

N. Correia and A. Barradas (emails: {ncorreia,abarra}@ualg.pt) are with the Faculty of Science and Technology, G. Schütz (email: gschutz@ualg.pt) is with Institute of Engineering, A. Mazayev and J. Martins are researchers (emails: {amazayev,jmartins}@ualg.pt, all at University of Algarve, 8005-139 Faro, Portugal.

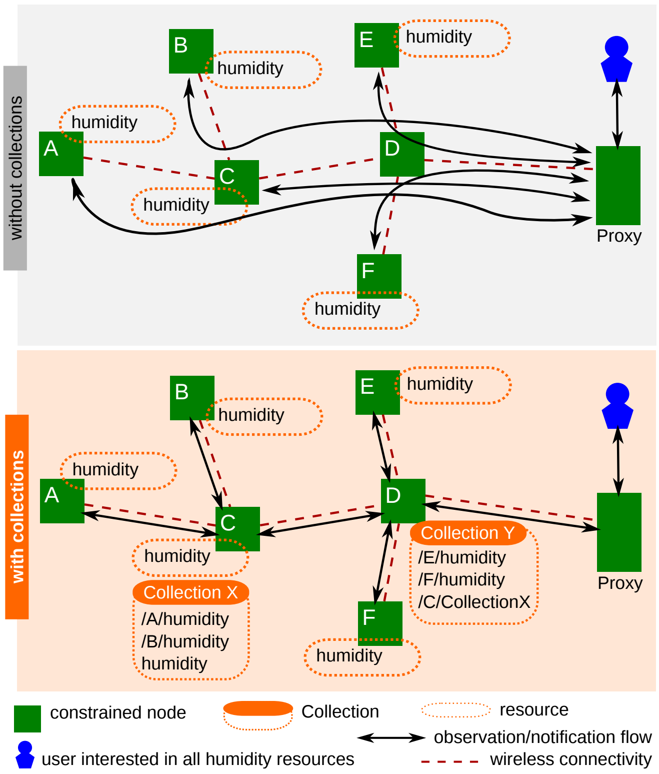

Fig. 1. Illustration of the need for collection resources.

Figure 1 illustrates the usage of Collection resources. As shown, when Collections are not available, there will be individual observation setups and individual notification flows between the constrained nodes and the proxy, and some congestion near the proxy is likely to happen. If Collections are built at the constrained nodes $C$ and $D$, the amount of signaling and notification packets at each link is significantly reduced, while also avoiding congestion near the proxy. Note that the observation of Collection resources is supported by CoAP and that notifications may include multiple observations, as stated in [4] and [7].

In this article, we propose the usage of CoRE Interfaces to build resource Collections with the purpose of saving network energy, allowing the lifetime of sensor networks to increase. The real challenge is to find the best Collection resources that should become available for observation by clients, leading to a balanced/fair energy consumption among nodes, while still considering that Collections may include other Collections. The main motivation behind fairness is that the depletion of a node will prevent clients from receiving data hosted in other servers, if such node is serving as a Collection/aggregation point.

In summary, the contributions of this article include:

- Establishing optimization procedures to determine the best set of Collections and observation steps throughout the network (using, if necessary, intermediate proxies), that evenly distribute energy consumption among nodes, increasing network lifetime. Collections may include other Collection resources besides atomic resources. An algorithm to find a feasible solution quickly is also proposed.
- Evaluation of the energy consumption and control/registration messages required when setting up or changing observations.
- Evaluation of the smallest number of Collections with a relevant impact on the overall energy consumption, allowing a reduction of control messages used for registration of observations, and consequently energy consumption.

We highlight that the resource discovery capabilities, together with the use of Interface Types to dynamically create/change resources, allow a full adaptation to any scenario while being a standards-based solution.

The remainder of this article is organized as follows. Section II presents CoRE-related standards and REST architecture, including the ongoing standardization on interfaces for resource design. Section III presents the related state of the art. The resource design problem is defined in Section IV, a mathematical formalization is presented in Section V, while a heuristic algorithm is proposed in Section VI. Performance evaluation is done in Section VII, while Section VIII draws conclusions.

## II. CoRE related standards

The discovery of resources is important for applications to run without human intervention. In HTTP, Web discovery and linking were specified in [8] and [9]. In the context of constrained nodes, resource discovery is referred to as *CoRE Resource Discovery* and is specified in [6].

### A. CoAP

While CoRE aims at realizing the REST architecture style in a suitable form for constrained nodes and networks, CoAP emerges as the Web application transfer protocol that has been designed for the special requirements of these constrained environments, especially considering energy, building automation, and other machine-to-machine (M2M) applications. CoAP provides a request/response interaction model between application endpoints. The "*coap*" and "*coaps*" URI schemes are used to identify CoAP resources and provide a mean to locate the resource.

For clients to be able to observe resources, and keep representations updated over time, the Observe extension to CoAP has been proposed in [4]. This extension allows the resource state at the client to be kept more consistent with the actual resource state at the server, and caches/proxies can be used for scalability. That is, an extended GET request can be either sent to a server, having such resource in its namespace, or to a proxy that is to be used as an intermediate [5],

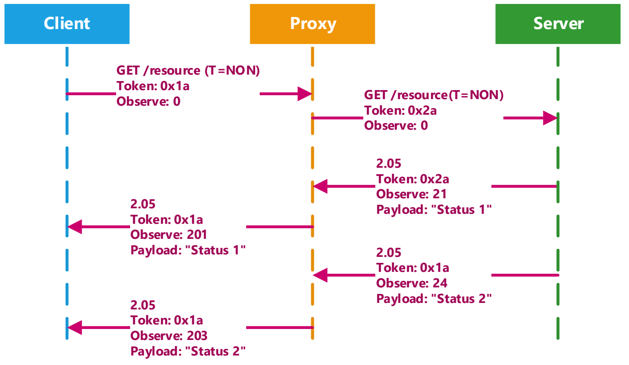

Fig. 2. Illustration of client registration through a proxy.

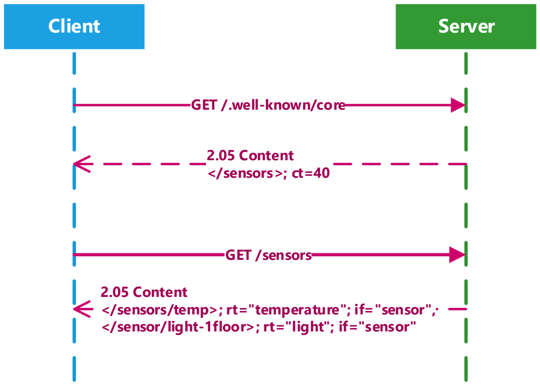
Client
Server
GET /.well-known/core
2.05 Content
</sensors>; ct=40
GET /sensors
2.05 Content
</sensors/temp>; rt="temperature"; if="sensor",
</sensor/light-1floor>; rt="light"; if="sensor"

Fig. 3. Resource discovery example.

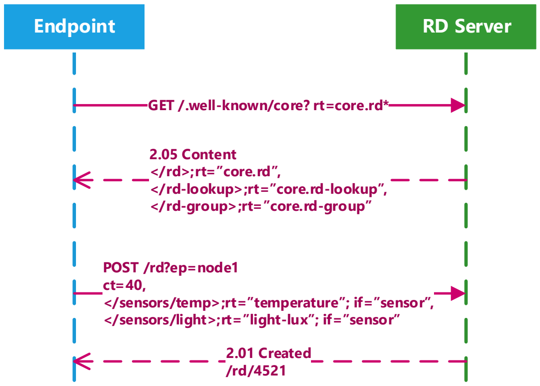
Endpoint
RD Server
GET /.well-known/core? rt=core.rd*
2.05 Content
</rd>;rt="core.rd",
</rd-lookup>;rt="core.rd-lookup",
</rd-group>;rt="core.rd-group"
POST /rd?ep=node1
ct=40,
</sensors/temp>;rt="temperature"; if="sensor",
</sensors/light>;rt="light-lux"; if="sensor"
2.01 Created
/rd/4521

Fig. 4. Registration at RD server.

[10]. The server, or proxy, will register the client as an observer, so that it can start receiving notifications, and responds with an extended response. Extended requests/responses are CoAP requests/responses with an Observe option inside them. This is illustrated in Figure 2. Multiple proxies in cascade can be used for scalability. This model allows notifications to be kept in cache until they expire, which is controlled by the *max-age* CoAP option. The "*Observe*:" option value is basically a sequence number associated with each response/notification, while the "*Token*:" option is used to match multiple notifications with a specific registration.

### B. CoRE Link Format

The default URN entry point ("*/.well-known/core*") has been specified in [6] for the discovery of resources hosted by a server. When a discovery is done, the URIs of the resources hosted by the server, attributes of resources, and relations are provided. The format of such entries is specified in [6] and is called CoRE Link Format. An IANA Media Type is assigned for CoRE Link Format payloads ("*application/link-format*").

Figure 3 shows a discovery example. Parameter "*ct*" indicates the Content-Format (e.g., 40 refers to the application/link-format), "*rt*" is used to assign an application-specific semantic type (e.g., "*temperature*", "*http://sweet.jpl.nasa.gov/2.0/phys.owl#temperature*") and "*if*" is used to specify an interface definition used to interact with the resource (e.g., "*sensor*", "*http://www.example.org/myapp.wadl#sensor*"). The definition of an interface allows every resource, method, request and response to be formally and precisely described.

### C. Resource Directory

CoAP provides a request/response interaction model between endpoints, supporting the discovery of resources, but direct discovery may not be practical, as nodes use sleeping modes to save energy. For this reason, a Resource Directory (RD) entity can be employed, as specified in [11]. This Web entity would host descriptions of resources held on various servers, allowing lookups from others. An RD supports Web interfaces for the discovery of the directory servers, registration/update/removal of resource descriptions, lookup of resources, and group maintenance. Figure 4 shows an RD discovery example, followed by a registration at the base URN "*/rd*". More specifically, endpoint "*node1*" registers "*</sensors/temp>*" and "*</sensors/light>*" resources at the RD server previously discovered. The response includes an RD generated location "*/rd/4521*".

### D. Resource Design Interfaces and Collections

A set of Interface Types for some resource design patterns has been recently proposed in [7], so that servers can compose/organize resources and clients may discover them. These resources can be of a Collection type.

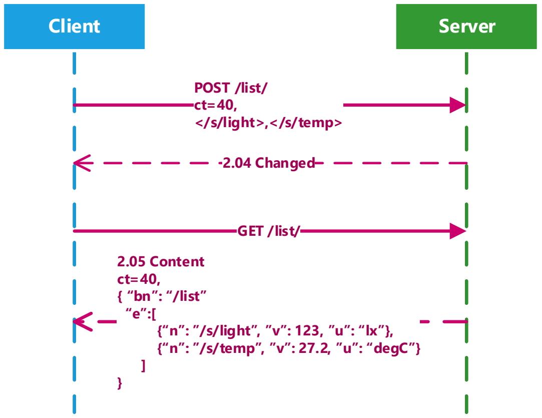

Fig. 5. Creation of Collection /list/.

*1) Collections:* A Collection is a resource representing one or more related resources. This allows resources to be organized for discovery, and for various forms of bulk interaction, according to the Interface Type used. An Interface Type defines the associated content format, data types, URI templates, REST methods, parameters and responses. The Interface Types discussed in [7] include: Link List, Batch, Linked Batch, Hypermedia Collection and Binding.

Collections may thus serve several purposes [7]:

- Provide gradual reveal of resources on an endpoint: the “*/.well-known/core*” URN would provide a small set of links pointing to Collections representing further clusters of resources (e.g., device information, device configuration, device management).
- Provide resource encapsulation: link embedding may be used to provide a single resource which a client interacts for obtaining a set of related resource values (e.g., single SenML data object)
.
- Group a set of similar resources for bulk state update or actuation: for example, a single update from a client could, in turn, update each resource of the Collection.

In this article, the purpose of using Collections is to increase network lifetime. That is, Collections should be built so that energy consumption is evenly distributed among nodes, while having the observation needs of clients in mind.

Resource links in CoRE Link-Format can be represented in several Content-Formats, including JSON, and resource values may be represented in SenML using a JSON variant, for example. In addition, a new *collection+senml+json* Content-Format is proposed in [7], if both need to be combined in a single representation. This is accomplished by extending SenML with a link element, “*l*”. For more details, see [7].

*2) Linked Batch Interface:* One of the Interface Types defined in [7] is the Linked Batch. This interface, specified by link parameter *if=“core.lb”*, allows the content of a Collection to change dynamically according to the control of a Web client. A request with a POST method and a CoRE Link Format content appends links to the Collection, a request with a PUT method allows the Collection to be updated, and DELETE removes the entire Collection. Let us assume that the following Collection needs to be created and filled, to be used as resource:

```
</list/>;rt="1stfloor";if="core.lb"
```

Figure 5 shows a POST to add two resources to Collection “</*list*/>”, while a GET obtains a single SenML data object including both resource values.

The possibility of building/updating Collections dynamically, using a Linked Batched Interface, together with a full knowledge of currently available resources (whose descriptions would be hosted in RDs), allows intelligent decisions to be done regarding the design of Collection resources and allows easy integration of such Collection resources into the network. In the present work, this is done with the objective of increasing network lifetime.

## III. State of the Art

Some work around CoAP started recently to emerge. In [12], the CoAP and HTTP are evaluated and compared in terms of energy consumption and response time. Concerning security, significant efforts have been done to use the *Datagram Transport Layer Security* (DTLS) to protect CoAP [13]. The usage of the authorization framework OAuth is also being investigated for IoT devices to access each other’s resources and grant specific authorization flows [3], [14].

Regarding the Observe extension of CoAP, a modification is proposed in [5], so that *Quality of Service* (QoS) is supported. The server classifies notifications into critical and non-critical and prioritizes their delivery. In [15], conditional observation is introduced. This is because many state changes are not relevant to clients, resulting in a waste of resources. With conditional observation, the client, instead of deciding whether to use a value or not, would be able to tell the server the criteria for notification. This is done through the insertion of a condition option that extends the Observe option.

Energy saving in CoAP-based constrained environments has been studied in [10] and [16]. In these articles, the multiple registration steps, of multiple observations, are planned so that notifications are aggregated at intermediate proxies, leading to increased energy saving and better bandwidth utilization. In [16], this point is explored for monitoring applications, and in [10] a multi-purpose framework is discussed that can be applied to multiple application scenarios. These articles use the notion of a transmission plan for

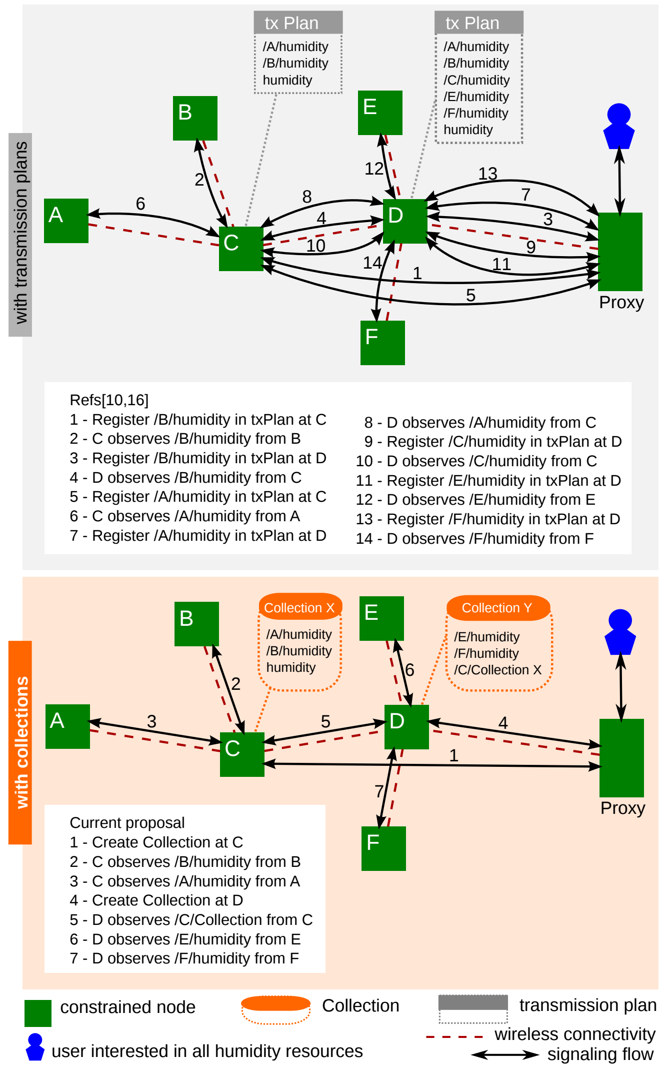
with transmission plans
tx Plan
/A/humidity
/B/humidity
humidity
tx Plan
/A/humidity
/B/humidity
/C/humidity
/E/humidity
/F/humidity
humidity
A
B
C
D
E
F
Proxy
Refs[10,16]
1 - Register /B/humidity in txPlan at C
2 - C observes /B/humidity from B
3 - Register /B/humidity in txPlan at D
4 - D observes /B/humidity from C
5 - Register /A/humidity in txPlan at C
6 - C observes /A/humidity from A
7 - Register /A/humidity in txPlan at D
8 - D observes /A/humidity from C
9 - Register /C/humidity in txPlan at D
10 - D observes /C/humidity from C
11 - Register /E/humidity in txPlan at D
12 - D observes /E/humidity from E
13 - Register /F/humidity in txPlan at D
14 - D observes /F/humidity from F
with collections
Collection X
/A/humidity
/B/humidity
humidity
Collection Y
/E/humidity
/F/humidity
/C/Collection X
Current proposal
1 - Create Collection at C
2 - C observes /B/humidity from B
3 - C observes /A/humidity from A
4 - Create Collection at D
5 - D observes /C/Collection from C
6 - D observes /E/humidity from E
7 - D observes /F/humidity from F
constrained node
Collection
transmission plan
user interested in all humidity resources
wireless connectivity
signaling flow

Fig. 6. Signaling comparison between proposed approach and previous work[10], [16].

aggregation to be achieved and, as far as we known, are the only ones to which the current proposal can be compared to, as they are also CoAP-based and aggregate resources for energy-saving purposes.

### *Comparison Between Proposed Approach and Previous Work [10], [16]*

The proposal in [10] and [16] is important to incorporate energy-saving concerns when planning registration steps for observation requests. However, no notion of Collection exists. That is, registration is done individually for each subject inside an aggregate. Therefore, the setup/change of registrations requires multiple individual messages (per subject) to be sent, increasing energy consumption. This drawback is caused by aggregates (transmission plans) in [10] and [16] not being available as observable resources, which happens with Collections. Another difference is that Collections may include both atomic resources and other Collections, and the aggregates in [10] and [16] can not incorporate other aggregates. These issues completely change the shape of the design problem that must be solved. To better identify differences, and the practical significance of the current approach, a comparison is done in Figure 6 regarding the scenario presented in Figure 1. More specifically, the signaling required to start monitoring all humidity sensors is compared. As shown, the current proposal significantly reduces signaling. The following should also be highlighted:

- Any change on observation requests, and consequent rearrangement of transmission plans, requires a lot of signaling in [10] and [16]. Many of such scenarios are easily addressed when Collections are used. For example, and still regarding the scenario at Figure 6, if the user wishes to monitor different sensors located at $A$ and $B$, then only the content of $C$'s Collection needs to change, together with the observations done by $C$. When using the approach in [10] and [16], all registrations/observations done at steps 1-8 would need to be canceled and new ones created.
- The approach in [10] and [16] does not ensure synchronized observations, meaning that aggregation of notifications might not occur. More specifically, if some notifications have a very low max-age value (see Section II-A) then such notifications can not wait for others and immediate aggregation is performed. Therefore, energy saving is not guaranteed. The use of Collections avoids this issue, as observation is performed on Collections and not on individual resources.

Section VII compares the performance of the proposed approach against the ones in [10] and [16], regarding the signaling overhead. Energy consumption associated with the forwarding of notifications is difficult to compare, as the algorithms in [10] and [16] are applied per arrival of observation request, while in the present work an overall optimization is being done considering all client observation requests. However, if the transmission plans in [10] and [16] were optimized for an overall set of observation requests, and all observations were syncronized (difficult to achieve as previously explained) then energy consumptions associated with notification flows (not signaling) would be similar to ours. That is, our approach corresponds to their best scenario (upper bound) regarding notification forwarding, assuming that fairness is ignored (unfair case analyzed in Section VII). Besides ensuring the best energy consumption scenario for the flow of notifications, the approach proposed here reduces signaling associated with observation setups reducing the energy consumption associated with those steps.

## IV. Resource Design Problem

The following notation is assumed when defining the resource design problem:

$\mathcal{N}$ Set of nodes in constrained network. Node $n \in \mathcal{N}$ can be a producer and/or consumer of resources.
$\mathcal{R}(n)$ Set of resources produced by $n \in \mathcal{N}$.
$\mathcal{S}$ Set of all subjects in $\mathcal{N}$.
$\mathcal{I}(n)$ Subject needs of resource consumer $n \in \mathcal{N}$.

A resource available at $n \in \mathcal{N}$ can be:

– a locally created/produced Atomic or Collection resource.
– in cache, being the outcome of an observation made to an Atomic or Collection resource produced in another node.

A Collection can include both locally produced and cached resources. For clarity, some definitions are presented next.

*Definition 1 (Producer):* A node $n \in \mathcal{N}$ is considered a producer if it has resources, of Atomic or Collection type, that are locally created/produced. The set $\mathcal{R}(n)$ denotes the set of resources produced at $n \in \mathcal{N}$, and $r$ is an element of such set.

Note that if an $r \in \mathcal{R}(n)$, $\forall n \in \mathcal{N}$, is of Collection type then it may incorporate Collection and/or Atomic resources produced in other nodes. Even so, $r$ is assumed to be produced in $n$.

*Definition 2 (Subject):* A subject is the matter (e.g., temperature) to which a resource can refer to. An Atomic resource relates to a single subject $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of all subjects available in $\mathcal{N}$. Different Atomic resources may relate to the same subject.

*Definition 3 (Consumer):* A resource consumer is any node $n \in \mathcal{N}$ that has a set of subject needs, denoted by $\mathcal{I}(n)$.

To fill the subject needs of resource consumers, a set of observations on resources (Atomic and Collection) will exist throughout the network, so that the notifications including the requested subjects flow towards resource consumers.

*Definition 4 (Observation):* A node $n \in \mathcal{N}$ is responsible for a set of observations on resources in $\cup_{n \in \mathcal{N}} \mathcal{R}(n)$. Node $n$ can be the resource consumer of resulting notifications, or just an intermediate/proxy node.

*Definition 5 (Resource Design Problem):* Given a constrained network including a set of nodes $\mathcal{N}$, each node $n \in \mathcal{N}$ being a producer of a set of Atomic resources and/or a consumer with subject needs/interests $\mathcal{I}(n)$, decide for: *i)* a Collection resource design at every node, increasing the number of resources available at the network; *ii)* a set of observations done by each node. Decisions should always try to maximize network lifetime (fair energy consumption among nodes) while meeting consumer's subject needs.

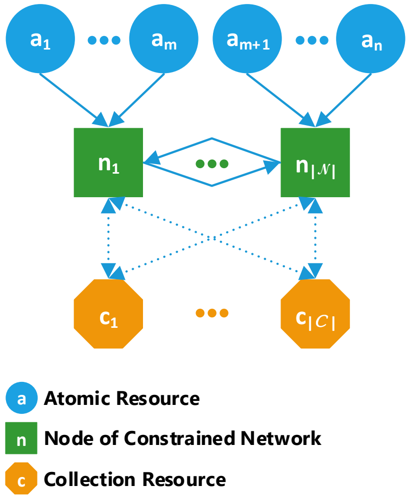

Fig. 7. Working graph incorporating virtual nodes and virtual edges.

## V. Mathematical Formalization

This section discusses the mathematical formalization of the resource design problem under study. For this purpose, a working graph is introduced that includes virtual Atomic nodes and virtual Collection nodes besides non-virtual nodes $\mathcal{N}$. Such working graph is shown in Figure 7. The following sections describe the extra notation and assumptions associated with it.

### A. Further Notation

The working graph in Figure 7 will be denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of vertices and edges, respectively. Regarding $\mathcal{V}$, this includes subsets $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{N}$. In summary:

$\mathcal{A}$ Virtual nodes representing Atomic resources ($\mathcal{A} \subset \mathcal{V}$). An $a \in \mathcal{A}$ produces a single subject: $\delta_s^a = 1$, if $a$ produces subject $s \in \mathcal{S}$; $\delta_s^a = 0$ otherwise.
$\mathcal{C}$ Virtual nodes representing Collection resources ($\mathcal{C} \subset \mathcal{V}$). Each $c \in \mathcal{C}$ will be filled with subjects flowing into it through edges.
$\mathcal{N}$ Non-virtual nodes ($\mathcal{N} \subset \mathcal{V}$). A node $n \in \mathcal{N}$ can be a consumer and/or proxy for Atomic and Collection resources.
$\mathcal{V}$ Set of nodes/vertices in the working graph ($\mathcal{V} = \mathcal{A} \cup \mathcal{C} \cup \mathcal{N}$).
$\mathcal{E}$ Set of edges in the working graph. The weight of an edge $e \in \mathcal{E}$ is denoted by $w(e)$.

Note that the number of Collections is limited, for resource management to be feasible. That is, after a certain number of Collections, the energy saving benefit might be too small compared to the overhead of managing more Collections. Such evaluation is done in Section VII. The following section discusses edges in $\mathcal{E}$ and how resources may flow through the working graph.

### *B. Edges and Observation Assumptions*

Edges between some types of nodes in $\mathcal{V}$ exist. The weight of an edge $e \in \mathcal{E}$ depends on the type of its endpoints, and is basically an energy cost associated with the flow of notifications from the source to the destination endpoint. The following edges exist:

*1) $e = (a, n)$, if $n \in \mathcal{N}$ is the producer of Atomic resource $a \in \mathcal{A}$:* The weight of such an edge $e$ will be $w(e) = 0$ since $a$ is connected to its producer.

*Assumption 1:* Any node $n \in \mathcal{N}$ when asked (observed) for a subject, may internally get such subject if it is the producer of an Atomic resource producing such subject. This has no impact on the energy consumption associated with the flow of notifications.

*2) $e = (c, n)$, $\forall c \in \mathcal{C}, \forall n \in \mathcal{N}$:* The weight of this edge will be:

$$w(e) = \begin{cases} 0, & \text{if } c \text{ is to be stored in node } n \\ \infty, & \text{otherwise} \end{cases} \tag{1}$$

Collection content, and their storage, are to be determined by the optimization model discussed next. Therefore, an edge from each Collection to every node must be made available. Their weights will change according to the Collection to node storage assignment. An $\infty$ weight will avoid the use of the edge.

*Assumption 2:* A node $n \in \mathcal{N}$, when asked (observed) for a subject, may get such subject from a Collection produced in it. This has no impact on the energy consumption associated with the flow of notifications.

*3) $e = (n_i, n_j)$, $\forall n_i, n_j \in \mathcal{N} : n_i \neq n_j$:* The weight of this edge $e$ will be $w(e) = OWD(n_i, n_j)$, where OWD is the one-way delay between the referred nodes. That is, the impact on energy will be proportional to the distance between nodes. For simplicity, the OWD is assumed to be the smallest number of hops between the nodes.

*Assumption 3:* Any node $n_i$, when asked (observed) for a subject, may get such subject from another node $n_j$ if it is not the producer of any resource (Atomic or Collection) including this subject. This increases the energy consumption associated with the flow of notifications.

*4) $e = (n, c)$, $\forall n \in \mathcal{N}, \forall c \in \mathcal{C}$:* The weight of such an edge $e$ will be:

$$w(e) = \begin{cases} 0, & \text{if } c \text{ is to be stored in node } n \\ \infty, & \text{otherwise} \end{cases} \tag{2}$$

Since Collection content, and their storage, are to be determined by the optimization problem, an edge from each node to every Collection must be made available. Their weights will change according to Collection to node storage assignment. An $\infty$ weight will avoid the use of the edge.

*Assumption 4:* A Collection node $c \in \mathcal{C}$ fills its content through the node $n \in \mathcal{N}$ hosting it (producing $c$). This node may request subjects from other nodes, according to the previous assumptions, which may flow individually or inside Collections.

This allows Collections to include other local resources (Atomic or Collections) or resources coming from other nodes (producers or proxies).

### *C. Objective and Constraints*

The following optimization problem determines the content and place of Collection resources, and finds a set of resource observations that meet consumer node interests. This is done while having the network lifetime increase as main goal. The variables required to find the best Collection resource design are:

| | |
|---|---|
| $\vartheta_s^{v_i,v_j}$ | Equal to one if $v_j \in \mathcal{N} \cup \mathcal{C}$ requests/observes an Atomic resource of subject $s \in \mathcal{S}$ from server node $v_i \in \mathcal{N} \cup \mathcal{A}$; zero otherwise. |
| $\chi_{s,c}^{n_i,v_j}$ | Equal to one if $v_j \in \mathcal{N} \cup \mathcal{C}$ requests/observes Collection resource $c \in \mathcal{C}$ from server node $n_i \in \mathcal{N}$, and such Collection provides subject $s \in \mathcal{S}$; zero otherwise. |
| $\eta_s^n$ | Equal to one if $n \in \mathcal{N}$ is requested/observed for Atomic resource of subject $s \in \mathcal{S}$; zero otherwise. |
| $\sigma_c^{n_i,n_j}$ | Equal to one if $n_j \in \mathcal{N}$ requests/observes Collection resource $c \in \mathcal{C}$ from server node $n_i \in \mathcal{N}$; zero otherwise. |
| $\rho_c^n$ | Equal to one if $n \in \mathcal{N}$ is requested/observed for Collection resource $c \in \mathcal{C}$; zero otherwise. |
| $\gamma_n^c$ | Equal to one if Collection $c \in \mathcal{C}$ is stored/produced at node $n \in \mathcal{N}$; zero otherwise. |
| $\zeta_{n_i}^A$ | Total cost associated with observation of Atomic resources at $n_i \in \mathcal{N}$. |
| $\zeta_{n_i}^C$ | Total cost associated with observation of Collection resources at $n_i \in \mathcal{N}$. |
| $\zeta^{UB}$ | Highest total cost ($\zeta^A$ plus $\zeta^C$) among all nodes, which should be minimized for fairness to be achieved. |

*1) Objective Function:*

$$OF^{\text{fair}}\text{: Minimize } \zeta^{UB} \tag{3}$$

*2) Constraints:* The previous objective function is constrained by the following:

– Flowing of subjects at Atomic resources:

$$\sum_{a_i\in\mathcal{A}}\sum_{n_j\in\mathcal{N}:e(a_i,n_j)\in\mathcal{E}}\vartheta_s^{a_i,n_j}\times\delta_s^{a_i}\geq 1, \\ ,\forall n_k\in\mathcal{N}:\mathcal{I}(n_k)\neq\emptyset,\forall s\in\mathcal{I}(n_k) \tag{4}$$

where $\delta_s^a$ is one if $a\in\mathcal{A}$ produces subject $s\in\mathcal{S}$, zero otherwise, which is known information.

$$\vartheta_s^{a_i,n_j}-\sum_{n_k\in\mathcal{N}}\vartheta_s^{n_j,n_k}=0,\forall n_j\in\mathcal{N}:\mathcal{I}(n_j)\neq\emptyset, \\ ,\forall s\in\mathcal{I}(n_j),a_i\in\mathcal{A}:e(a_i,n_j)\in\mathcal{E} \tag{5}$$

$$\eta_s^{n_i}\geq\vartheta_s^{n_i,v_j},\forall n_i\in\mathcal{N},\forall v_j\in\mathcal{N}\cup\mathcal{C},\forall s\in\mathcal{S} \tag{6}$$

$$\sum_{a_i\in\mathcal{A}:e(a_i,n_j)\in\mathcal{E}}\vartheta_s^{a_i,n_j}+\sum_{n_i\in\mathcal{N}:e(n_i,n_j)\in\mathcal{E}}\vartheta_s^{n_i,n_j}=\eta_s^{n_j}, \\ ,\forall n_j\in\mathcal{N},\forall s\in\mathcal{S} \tag{7}$$

$$\sum_{a_i\in\mathcal{A}:e(a_i,n_j)\in\mathcal{E}}\vartheta_s^{a_i,n_j}+\sum_{n_i\in\mathcal{N}:e(n_i,n_j)\in\mathcal{E}}\vartheta_s^{n_i,n_j}\geq\vartheta_s^{n_j,c}, \\ ,\forall n_j\in\mathcal{N},\forall c\in\mathcal{C},\forall s\in\mathcal{S} \tag{8}$$

The constraints (4) force requested subjects to emanate from nodes producing Atomic resources, while (5) ensure flow conservation through other nodes. Constraints (6) fill variables that indicate if a node $n_i\in\mathcal{N}$ is being requested/observed for subject $s\in\mathcal{S}$, which happens if a non-virtual node or Collection node requests it. Constraints (7) state that if a node $n_j\in\mathcal{N}$ is requested for subject $s\in\mathcal{S}$ then it will get it either locally (from Atomic node connected to it) or will request/observe such subject from another node in $\mathcal{N}$. These comply with Assumptions 1−4. Finally, Constraints (8) force nodes to request subjects from Atomic resources or other nodes when these are requested by Collections.

– Flowing of Collections:

$$\rho_{c_k}^{n_i}\geq\sigma_{c_k}^{n_i,n_j},\forall n_i,n_j\in\mathcal{N},\forall c_k\in\mathcal{C} \tag{9}$$

$$\rho_{c_k}^{n_i}\geq\chi_{s,c_k}^{n_i,c_j},\forall n_i\in\mathcal{N},\forall c_j,c_k\in\mathcal{C}:c_j\neq c_k,\forall s\in\mathcal{S} \tag{10}$$

$$\gamma_{n_j}^{c}+\sum_{n_i\in\mathcal{N}:e(n_i,n_j)\in\mathcal{E}}\sigma_c^{n_i,n_j}\geq\rho_c^{n_j},\forall n_j\in\mathcal{N},\forall c\in\mathcal{C} \tag{11}$$

The constraints (9) and (10) will fill variables that indicate if a node $n_i\in\mathcal{N}$ is being requested/observed for a Collection resource $c_k\in\mathcal{C}$, which happens if a node $n_j\in\mathcal{N}$ or Collection $c_j\in\mathcal{C}$ requests it. In the last case, this means that Collections may include other Collections. Constraints (11) state that if a node $n_j$ is requested/observed for a Collection resource then it must get it either locally or from any other node $n_i\in\mathcal{N}$.

– Subjects supplied by Collections:

$$\chi_{s,c_k}^{n_i,n_j}\geq\sigma_{c_k}^{n_i,n_j}+\sum_{n_l\in\mathcal{N}}\vartheta_s^{n_l,c_k}-1,\forall n_i,n_j\in\mathcal{N}, \\ ,\forall c_k\in\mathcal{C},\forall s\in\mathcal{S} \tag{12}$$

$$\chi_{s,c_k}^{n_i,n_j}\geq\sigma_{c_k}^{n_i,n_j}+\sum_{n_l\in\mathcal{N}}\sum_{c_m\in\mathcal{C}}\chi_{s,c_m}^{n_l,c_k}-1,\forall n_i,n_j\in\mathcal{N}, \\ \forall c_k\in\mathcal{C},\forall s\in\mathcal{S} \tag{13}$$

These Constraints force all subjects inside a Collection to flow inside nodes requesting it. The next constraints are required to ensure that subjects are not supplied through a Collection if such collection is not requested, or if such Collection does not include the subject.

$$\chi_{s,c_k}^{n_i,n_j}\leq\sigma_{c_k}^{n_i,n_j},\forall n_i\in\mathcal{N},\forall n_j\in\mathcal{N},\forall c_k\in\mathcal{C},\forall s\in\mathcal{S} \tag{14}$$

$$\chi_{s,c_k}^{n_i,n_j}\leq\sum_{n_l\in\mathcal{N}}\vartheta_s^{n_l,c_k}+\sum_{n_l\in\mathcal{N}}\sum_{c_m\in\mathcal{C}}\chi_{s,c_m}^{n_l,c_k},\forall n_i\in\mathcal{N}, \\ ,\forall n_j\in\mathcal{N},\forall c_k\in\mathcal{C},\forall s\in\mathcal{S} \tag{15}$$

– Hosting of Collections:

$$\sum_{n\in\mathcal{N}}\gamma_n^{c}=1,\forall c\in\mathcal{C} \tag{16}$$

$$\sum_{s\in\mathcal{S}}\vartheta_s^{n_i,c_j}+\sum_{c_k\in\mathcal{C}}\sum_{s\in\mathcal{S}}\chi_{s,c_k}^{n_i,c_j}\leq\gamma_{n_i}^{c_j}\times\Delta,\forall n_i\in\mathcal{N}, \\ ,\forall c_j\in\mathcal{C} \tag{17}$$

As stated in Assumption 4, Collections should only get subjects from the node hosting them, which is determined by Constraints (16). To prevent Collections from getting subjects from other nodes Constraints (17) are used, where $\Delta$ is a big value. This complies with Assumption 4.

– Subjects supplied by Atomic nodes:

$$\sum_{n_j\in\mathcal{N}:e(a_i,n_j)\in\mathcal{E}}\vartheta_s^{a_i,n_j}\leq\delta_s^{a_i},\forall a_i\in\mathcal{A},\forall s\in\mathcal{S} \tag{18}$$

– Filling interests of final consumers:

$$\sum_{n_i\in\mathcal{N}:n_i\neq n_j}\vartheta_s^{n_i,n_j}+\sum_{n_i\in\mathcal{N}:n_i\neq n_j}\sum_{c\in\mathcal{C}}\chi_{s,c}^{n_i,n_j}\geq 1, \\ ,\forall n_j\in\mathcal{N}:\mathcal{I}(n_j)\neq\emptyset,\forall s\in\mathcal{I}(n_j) \tag{19}$$

These state that a node $n_j \in \mathcal{N}$ with a subject interest may request/observe such subject as an Atomic resource from another node in $\mathcal{N}$, or may request/observe a Collection including such subject. In both cases, the node being observed/requested can be the resource producer or an intermediate/proxy.

– Cost associated with the flow of Atomic resources:

$$\zeta^A_{n_i} = \sum_{n_j \in \mathcal{N}} \sum_{s \in \mathcal{S}} \vartheta_s^{n_i,n_j} \times OWD(n_i,n_j), \forall n_i \in \mathcal{N} \quad (20)$$

– Cost associated with the flow of Collection resources:

$$\zeta^C_{n_i} = \sum_{n_j \in \mathcal{N}} \sum_{c \in \mathcal{C}} [(\sum_{s \in \mathcal{S}} \chi_{s,c}^{n_i,n_j} \times \Phi^1) \times OWD(n_i,n_j)] + \\ +(\sigma_c^{n_i,n_j} \times \Phi^2) \times OWD(n_i,n_j), \forall n_i \in \mathcal{N} \quad (21)$$

where $\Phi^1$ is a subject data size to standard notification size ratio, while $\Phi^2$ is a Collection notification constant envelop size to standard notification size ratio. The standard notification size is an average size of Atomic resource notification.

– Lowest total cost:

$$\zeta^{UB} \geq \zeta^A_{n_i} + \zeta^C_{n_i}, n_i \in \mathcal{N} \quad (22)$$

– Non-negativity assignment to variables:

$$\vartheta_s^{v_i,v_j}, \chi_{s,c}^{n_i,v_j}, \eta_s^n, \sigma_c^{n_i,n_j}, \rho_c^n, \gamma_n^c \in [0,1]; \zeta^A_{n_i}, \zeta^C_{n_i}, \zeta^{UB} \in \Re^+. \quad (23)$$

The CPLEX[1] optimizer was used to solve this problem, and the solution found will be the optimal solution for the resource design problem under consideration. However, this is a *Mixed Integer Linear Programming* (MILP) problem and these are generally NP-hard. Therefore, CPLEX will take a long time for large instances of the problem. For this reason a heuristic algorithm is proposed in the following section.

## VI. Heuristic Algorithm

The algorithm proposed to design Collection resources, while having network lifetime increase as main goal, has two steps:

I) *Building an Initial Working Graph*: An initial working graph including just Atomic and non-virtual nodes, $\mathcal{A}$ and $\mathcal{N}$, which is known information, is expanded to include a population of Collections.

```
$\mathcal{C} = \emptyset$;
repeat
    /* find potential Collections */;
    $\Pi = \{(r_i, r_j) : \{r_i \cup r_j\} \in \mathcal{I}(n_k), n_k \in \mathcal{N}$ and $r_i, r_j \in \mathcal{A} \cup \mathcal{C}\}$;
    for each $(r_i, r_j) \in \Pi$ do
        /* find nodes hosting $r_i$ and $r_j$ */;
        $n_i = \{n \in \mathcal{N} : (r_i, n) \in \mathcal{E}\}$;
        $n_j = \{n \in \mathcal{N} : (r_j, n) \in \mathcal{E}\}$;
        /* define nodes interested in $(r_i, r_j)$ */;
        $\Gamma = \{n \in \mathcal{N} : \{r_i \cup r_j\} \subset \mathcal{I}(n)\}$;
        /* see if Collection brings energy benefits */;
        for each $n_k \in \Gamma$ do
            $w_1 = \Phi^1 \times size(r_i)$;
            $w_2 = \Phi^1 \times size(r_j)$;
            $w_3 = \Phi^2 + (\Phi^1 \times size(r_i \cup r_j))$;
            if $OWD(n_i, n_j) \times w_1 + OWD(n_j, n_k) \times w_3 < OWD(n_i, n_k) \times w_1 + OWD(n_j, n_k) \times w_2$ then
                $c = \{r_i \cup r_j\}$;
                $label(c) = OWD(n_i, n_j) \times w_1 + OWD(n_j, n_k) \times w_3$;
                /* insert Collection in working graph */;
                $\mathcal{C} \leftarrow c$;
                /* insert corresponding edges in working graph */;
                $\mathcal{E} \leftarrow (c, n_j)$;
                $\mathcal{E} \leftarrow (n_j, c)$;
                break;
            end
        end
    end
until working graph does not change ;
```

II) *Pruning Working Graph*: It is necessary to remove Collections considering their usefulness and fairness in energy consumption among nodes, until the number of allowed Collections is reached.

```
for each $n \in \mathcal{N}$ do
    /* determine cost of each Collection in $n$ */;
    for each $c \in \mathcal{C}$ such that $(c, n) \in \mathcal{E}$ do
        /* determine clients not interested in $c$ */;
        $\Theta = \{n \in \mathcal{N} : c \not\subset \mathcal{I}(n)\}$;
        $cost(c) = label(c) \times |\Theta|$;
    end
end
repeat
    /* pick node with highest total Collection cost */;
    $n^* = argmax_{n \in \mathcal{N}} \{\sum_{c \in \mathcal{C}:(c,n) \in \mathcal{E}} cost(c)\}$;
    /* remove highest cost Collection from $n^*$ */;
    $c^* = argmax_{c \in \mathcal{C}:(c,n^*) \in \mathcal{E}} \{cost(c)\}$;
    $\mathcal{C} = \mathcal{C} \backslash \{c^*\}$;
until $|\mathcal{C}| = number\ of\ allowed\ Collections$ ;
```

The above procedures define which Collections will be available at the constrained network, specifying the values for variables $\gamma_n^c$ in the mathematical formalization of Section V. This significantly reduces the problem size, as many $\gamma_n^c$ variables will be set to zero, meaning that all related variables will also be set to zero (e.g., $\vartheta_s^{v_i,v_j}$ and $\chi_{s,c}^{n_i,v_j}$) in Constraints (17). This affects all constraints, as the space of feasible solutions becomes much smaller. Note that the CPLEX optimizer might choose to leave some Collections empty if these are not profitable for the goal that has been defined.

To determine the computational complexity of the

[1] IBM ILOG CPLEX Optimizer.

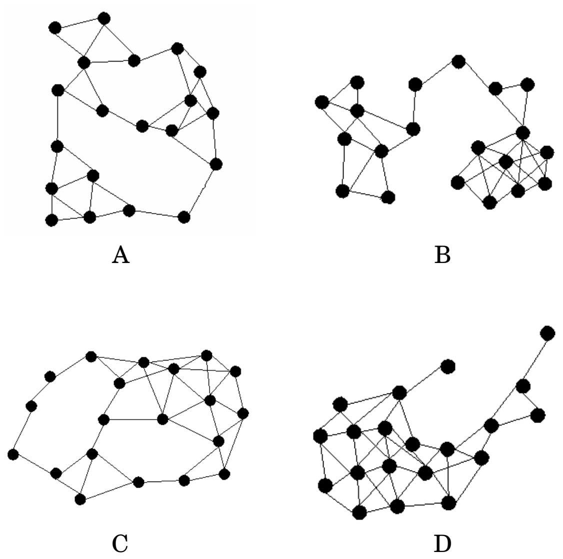

Fig. 8. Network topologies under test.

heuristic algorithm, we must be aware that the cycle at line 2 of step I will be performed at the worst case $|\bar{\mathcal{I}}|$ times, where $|\bar{\mathcal{I}}|$ is assumed to be the maximum number of subject interests of any consumer. After this number of steps the working graph does not change, and the procedure stops. Determining $\Pi$ at line 4 of step I requires at the worst case $|\mathcal{N}| \times \frac{|\mathcal{A}|^{2\times(|\bar{\mathcal{I}}|-1)}}{2}$ operations, as combinations of resources, which increase on each round because Collections are being built, must be searched to see if they fit consumer interests. This is also the worst case scenario for the remaining operations of step I. Therefore, the overall complexity of step I will be $O(|\bar{\mathcal{I}}| \times |\mathcal{N}| \times |\mathcal{A}|^{2\times(|\bar{\mathcal{I}}|-1)})$. Step II has a similar complexity, as non-virtual nodes, potential Collections and interests will be analyzed for pruning. If the maximum number of subject interests at consumers is low, then the complexity can be considered reasonable, given the mathematical problem under analysis.

## VII. Performance Analysis

### A. Network Setup

The network topologies under test have 20 sensor nodes and are the ones shown in Figure 8. These are randomly generated graphs using the weighted proximity algorithm described in [17]. It is assumed that each node $n \in \mathcal{N}$ produces a single Atomic resource. The subjects of Atomic resources are assumed to be different throughout the network. The number of consumer nodes, with interests, is assumed to be a quarter of the total number of nodes, and these are randomly selected. The average number of subject interests requested by each consumer is 10% of the total number of nodes. To determine the subject interests of nodes, each network is assumed to be divided into four quadrants. When a node is selected to become a consumer, it will select its subject interests either randomly or from the set of subject interests of its neighbours (same quadrant) meaning that neighbouring nodes will have similar interests. This is so because nearby nodes are usually interested in observing similar resources.

TABLE I
Network Parameters.

| Parameter | Value |
|---|---|
| Bandwidth ($B$) | 250000 bits=31250 bytes |
| Byte time ($T^{\min}$) | $\frac{1}{B} = \frac{1}{31250}$s |
| Duty cycle ($D$) | $\frac{0.125s}{T^{\min}}$ |
| Avg notification size ($L$) | 50 bytes |
| Subject data to notification size ratio ($\Phi^1$) | 0.5 |
| Collection envelop to notification size ratio ($\Phi^2$) | 0.5 |

The total energy consumption resulting from transmission of notifications, and the maximum energy consumption among nodes, are analyzed for both random and similar interests. For each such scenario, a different number of allowed collections is tested.

### B. Energy Consumption Estimation

Table I summarizes the remaining network parameters, based on [15] and [19], assuming 802.15.4 devices and contikiMAC *Low Power Listening* (LPL). This table includes the definition of byte time, or time for a byte to be transmitted, which is required to calculate an energy consumption. The devices are assumed to use a listening protocol with a duty cycle of 0.125s (value for contikiMAC LPL), as in [15], which leads to a duty cycle in byte times equal to $D = \frac{0.125\text{s}}{T^{\min}} = 3906.25$. This duty cycle must be considered when accounting for energy consumption.

When calculating an energy consumption, it is assumed that nodes use 24mW in awake mode and 52.2mW when transmitting/receiving; these are values for Z1 motes taken from [15]. Thus, the following energy consumption values (per byte time) are adopted:

$$E^T = 167.04 \times 10^{-8}\text{J}$$
$$E^W = 76.8 \times 10^{-8}\text{J}$$

We also assume that:

- Notification packets that are too large to fit into a single 802.15.4 frame are fragmented since an *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN) stack is assumed (see [20]).
- Considering a specific $n_j \in \mathcal{N}$, all notification transmissions originating from $n_j$ are included in the same duty cycle or in adjacent duty cycles.

The CPLEX optimizer is used to find the optimal solution for each specific resource design problem, and to determine the best values for variables. Then, considering a specific node $n_i$, the number of byte times for transmission of notifications is calculated using:

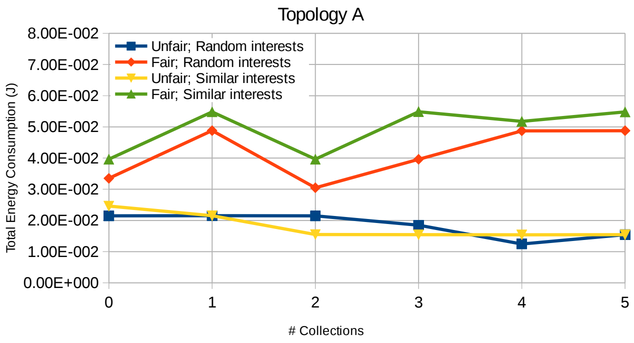

Fig. 9. Topology A: total energy consumption.

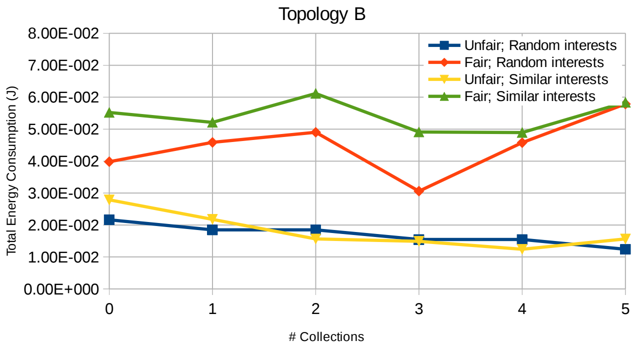

Fig. 10. Topology B: total energy consumption.

$$BT^T(n_i) = L \times \sum_{n_j \in \mathcal{N}} \sum_{s \in \mathcal{S}} \vartheta_s^{n_i,n_j} +$$

$$+L \times [\sum_{n_j \in \mathcal{N}} \sum_{c \in \mathcal{C}} (\sum_{s \in \mathcal{S}} \chi_{s,c}^{n_i,n_j} \times \Phi^1) + \sigma_c^{n_i,n_j} \times \Phi^2]. \quad (24)$$

The number of byte times in awake mode will be:

$$BT^W(n_i) = \lceil \frac{BT^T(n_i)}{D} \rceil \times D - BT^T(n_i), \quad (25)$$

and the total energy consumption resulting from transmission of notifications will be:

$$E^{\text{Total}} = \sum_{n_i \in \mathcal{N}} [(BT^T(n_i) \times E^T) + (BT^W(n_i) \times E^W)]. \quad (26)$$

Note that these values relate to energy consumed by CoAP/Observe nodes when transmitting notifications towards consumer nodes, either directly or through proxy nodes. That is, no forwarding at the routing layer is accounted.

### C. Fair vs Unfair Resource Design

To better evaluate the impact of the fair resource design approach, when compared with an unfair approach, a second objective function is also implemented. This second objective function has no fairness into consideration and chooses to reduce the overall energy consumption, and is defined as follows:

$$OF^{\text{unfair}}\text{: Minimize} \sum_{n_i \in \mathcal{N}} \zeta_{n_i}^A + \zeta_{n_i}^C. \quad (27)$$

Results for both $OF^{\text{fair}}$ and $OF^{\text{unfair}}$ (Eq. (3) and (27), respectively) are obtained for comparison.

### D. Results on Total Energy Consumption

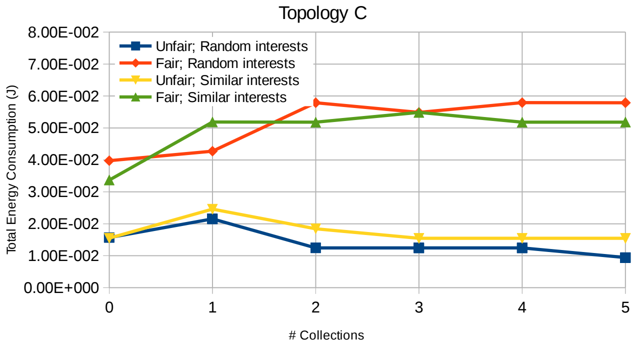

Fig. 11. Topology C: total energy consumption.

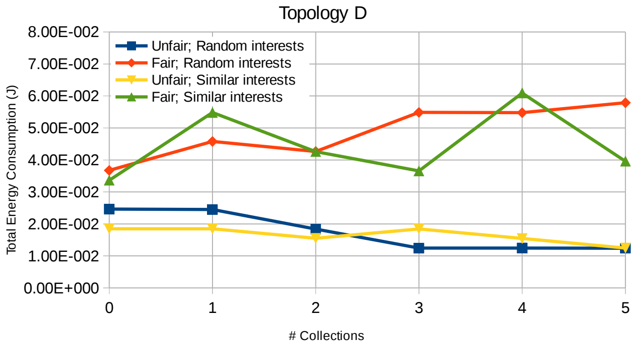

Fig. 12. Topology D: total energy consumption.

Figures 9–12 show the total energy consumption for topologies A, B, C and D, respectively. As previously said, these relate to energy consumed by CoAP/Observe nodes when transmitting notifications towards consumer nodes, either directly or through proxy nodes, and it is assumed that registrations, for observations to occur, have been previously done. Values obtained for fair and unfair objective functions are both shown in the cases of random and similar interests.

From plots, it is possible to observe that the unfair approach is able to get lower total energy consumption values than the fair approach, for both scenarios of random and similar interests. For the unfair approach, the total energy consumption decreases as the number of Collections increases, while for the fair approach it

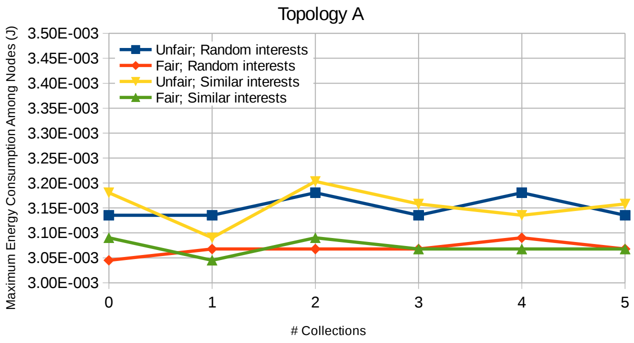

Fig. 13. Topology A: maximum energy consumption among nodes.

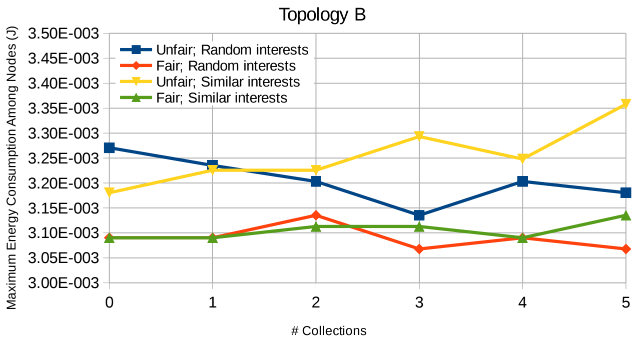

Fig. 14. Topology B: maximum energy consumption among nodes.

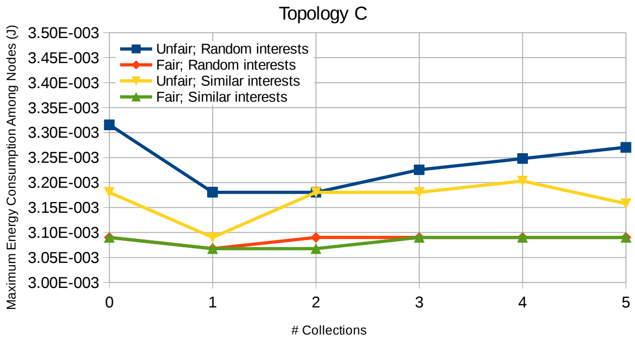

Fig. 15. Topology C: maximum energy consumption among nodes.

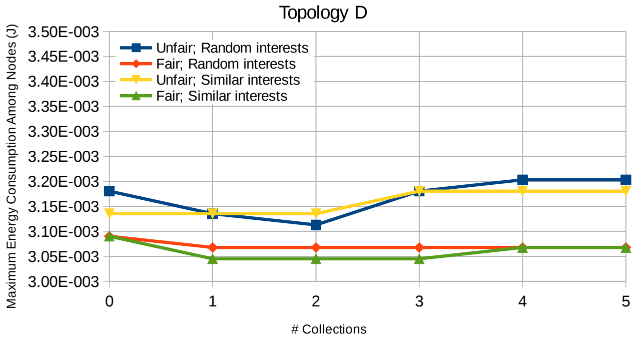

Fig. 16. Topology D: maximum energy consumption among nodes.

oscillates. Note that the optimization problem takes the OWD into consideration, for the minimization of global network energy consumption, and energy plots show the energy consumed at CoAP/Observe nodes where registrations exist and notifications need to be transmitted, which does not include nodes that make forwarding at the routing layer. CoAP/Observe nodes are the ones involved in registrations and, therefore, are flow concentration points with increased depletion probability. For this reason, we focus on them. This slight difference between the goal of the optimization problem and energy accounting leads to some oscillations in total energy values obtained (i.e for the unfair case an increase in available Collections does not always decrease the energy values accounted by Eq. (26), although global network energy consumption has decreased due to the minimization goal). However, the global trend is for energy consumption to decrease as the number of available Collections increases, for the unfair approach. Topology C presents energy consumption values that always increase for the fair case, as there is a network section with no alternative paths, becoming a congestion point.

When comparing the scenarios of random and similar interests, it seems that similar interests lead to more energy consumption in less evenly connected networks due to the non-existence of alternative paths, and this effect is stronger for a small number of Collections. More Collections can balance this effect. This can be confirmed by the plots in Figures 13−16, analyzed in the following section, where Topology B presents higher values in the scenario of similar interests.

### *E. Results on Maximum Energy Consumption Among Nodes*

Figures 13−16 show the maximum energy consumption among CoAP/Observe nodes transmitting notifications. That is, $max_{n_i \in \mathcal{N}}(BT^T(n_i) \times E^T + BT^W(n_i) \times E^W)$. It is possible to see that the unfair approach presents higher values, meaning that such approach is able to reduce the total energy consumption (Figures 9−12) at the expense of depleting some nodes. The fair approach has lower maximums, especially for evenly connected networks. For the fair approach, no particular difference exists between the scenarios of random or similar interests, meaning that an appropriate Collection placement is able to help minimize the maximum energy consumption, regardless of consumer interests.

### *F. Results on Total Number of Registration Msgs*

In regard to the number of registration messages required by the approach being proposed (aggregation of notifications using Collections), and the approach

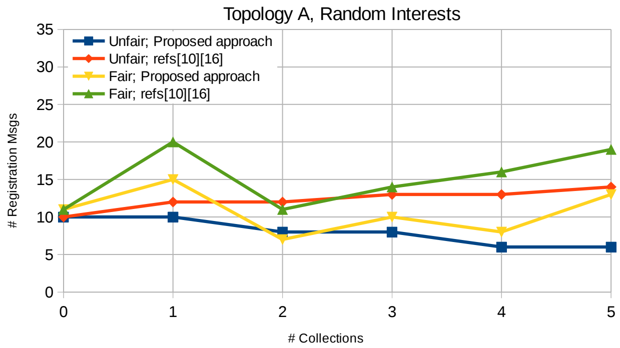

Fig. 17. Topology A: # of registration messages for random interests.

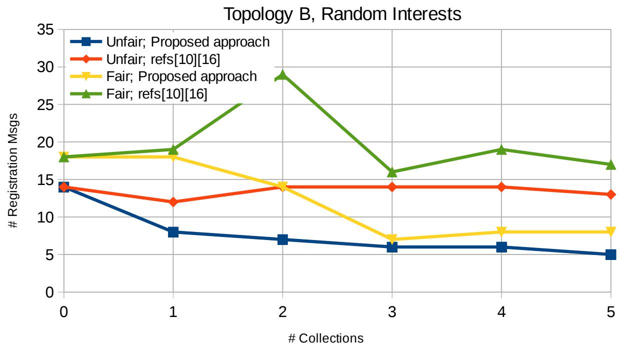

Fig. 18. Topology B: # of registration messages for random interests.

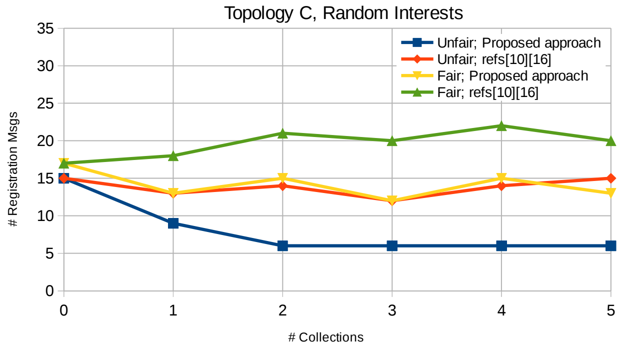

Fig. 19. Topology C: # of registration messages for random interests.

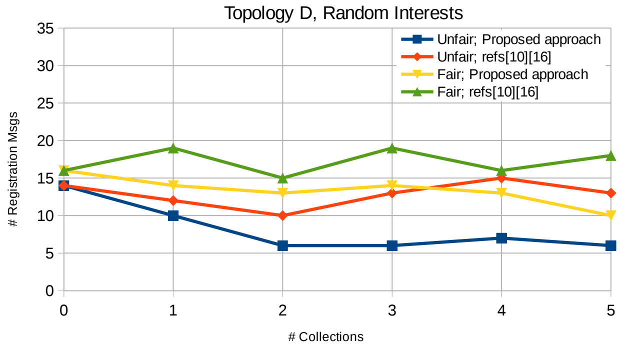

Fig. 20. Topology D: # of registration messages for random interests.

in [10] and [16] (aggregation of notifications with no Collections), Figures 17−20 show results for the scenario of random interests, while Figures 21−24 show results for the case of similar interests. To obtain these values the non-zero variables $\vartheta_s^{v_i,v_j}$ and $\chi_{s,c}^{n_i,v_j}$ were counted for the approach in [10], [16], while the $\vartheta_s^{v_i,v_j}$ and $\sigma_c^{n_i,n_j}$ were counted for the proposed approach (use of Collections). That is, in [10], [16] the registration would have to be done individually for each subject, while in our case registrations already incorporate Collections (there is no need for the individual registration of subjects included in the Collection).

Results show that the number of registration messages is much lower when Collections are used, and that difference increases for a higher number of available Collections. A such number of registration messages has also impact on energy consumption, although at the observation's set up phase and not at the notification's transmission phase. Since the proposed approach consumes less energy for observation set up, optimization of the network can be done more frequently in case client observation interests change, and Collection placement and content need to be changed.

Regarding the fair and unfair approaches, we can observe that more registration messages are required by the fair approach. This means that the fair approach increases the network lifetime by delaying node depletion, according to plots in Figures 13−16, during the notification transmission period (after the reconfiguration phase), although during setup of observations more registration messages are sent when compared to the unfair approach. Therefore, how frequent observation setup reconfiguration is done is a relevant issue. However, for both unfair and fair approaches, using Collections significantly reduces the number of registration messages.

In general, less evenly connected networks show bigger differences between the number of registration messages, when compared with evenly connected networks, mainly because they require more registration steps through intermediate nodes. This happens for the proposed approach and also the one in [10], [16].

Please note that during the transmission of notifications, the approach in [10], [16] would provide values similar to the unfair approach proposed here because optimal aggregation is also performed. For this reason, these were not plotted. However, the approach in [10], [16] does not provide fairness, putting into question network lifetime, and ends up consuming more energy during the observation setup phase since more registration messages are required. Therefore, the use of CoRE Interfaces and Collections brings energy benefits.

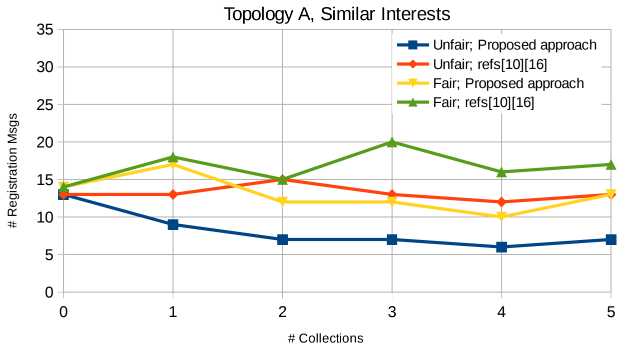

Fig. 21. Topology A: # of registration messages for similar interests.

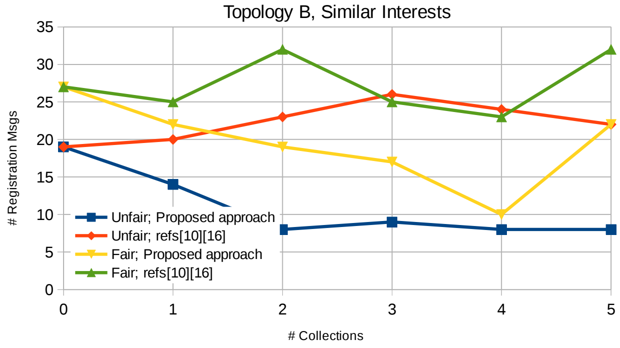

Fig. 22. Topology B: # of registration messages for similar interests.

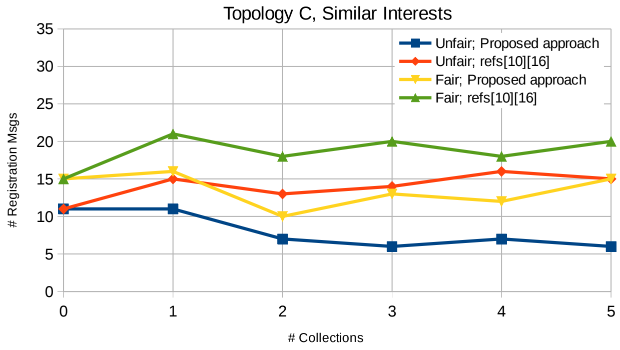

Fig. 23. Topology C: # of registration messages for similar interests.

### G. Heuristic Algorithm Performance

As previously stated, finding the optimal solution for the resource design problems under consideration might take too long if instances are large and, for this reason, an heuristic algorithm was proposed. Heuristic approaches employ some method for solutions to be obtained faster, but optimality is not ensured.

When comparing the solutions provided by the heuristic proposed in Section VI, against the optimal ones provided by $OF^{\text{fair}}$, we observed that the heuristic is able to choose Collections that are productive, reaching the optimal many times. More specifically, for most

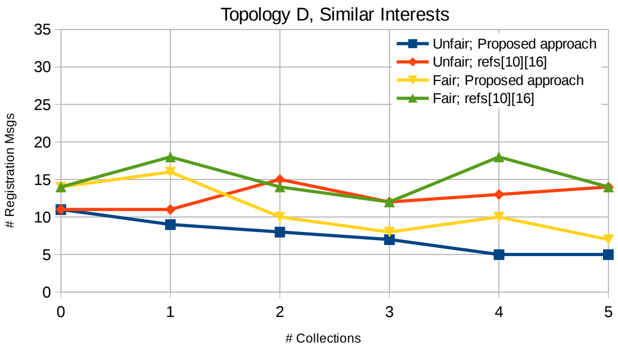

Fig. 24. Topology D: # of registration messages for similar interests.

connected networks (topologies C and D) optimality is always reached, for any number of Collections, while for less connected networks (topologies A and B) this happens $\backsim$ 80% of the times[2]. Note, however, that $OF^{\text{fair}}$ minimizes an upper bound and in such cases it becomes easier for algorithms to approach the optimal case.

When using the heuristic algorithm, to specify the $\gamma_n^c$ values in the mathematical formalization, runtimes decreased to values around a few seconds, instead of the minutes and hours associated with the optimal solution.

## VIII. Conclusion

In this article, CoRE Interfaces and Collections are used to reduce energy consumption on constrained sensor networks. An optimization procedure is proposed to determine the best set of Collections and observation steps that evenly distribute energy consumption across the network, so that the total network lifetime increases. This approach proved to be energy efficient, and when compared with previous approaches fewer registration messages are required to set up or change observations. This means that the network can be kept optimized while client observation interests change.

## Acknowledgment

This work was supported by the FCT (Foundation for Science and Technology) from Portugal, and developed within the CEOT (Center for Electronic, Optoelectronic and Telecommunications) and UID/MULTI/00631/2013 project.

## References

[1] L. Atzoria, A. Ierab and G. Morabito: The Internet of Things: A Survey, Computer Networks, Elsevier, Vol. 54, No. , 2010.

[2] Yao-Chung Hsu, Chi-Han Lin and Wen-Tsuen Chen: Design of a Sensing Service Architecture for Internet of Things with Semantic Sensor Selection, IEEE 11th International Conference on Ubiquitous Intelligence and Computing, December 2014.

[2] Plots are not shown because a significant number of values would be overlapped.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2678517, IEEE Internet of Things Journal

[3] Sye Loong Keoh, Sandeep S. Kumar, and Hannes Tschofenig: Securing the Internet of Things: A Standardization Perspective, IEEE Internet of Things Journal, May 2014.

[4] K. Hartke: ”Observing Resources in CoAP”, draft-ietf-core-observe-16, IETF, 2014.

[5] A. Ludovici, E. Garcia X. Gimeno and A. Calveras Auge: ”Adding QoS support for timeliness to the observe extension of CoAP”, IEEE Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2012.

[6] Z. Shelby: ”Constrained RESTful Environments (CoRE) Link Format”, RFC 6690, 2012.

[7] Z. Shelby, M. Vial, M. Koster and C. Groves: ”Reusable Interface Definitions for Constrained RESTful Environments”, draft-ietf-core-interfaces-07, 2016.

[8] M. Nottingham: ”Web Linking”, RFC 5988, 2010.

[9] M. Nottingham and E. Hammer-Lahav: ”Defining Well-Known Uniform Resource Identifiers (URIs)”, RFC 5785, 2010.

[10] N. Correia, D. Sacramento and G. Schütz: ”Dynamic Aggregation and Scheduling in CoAP/Observe Based Wireless Sensor Networks”, IEEE Internet of Things, Vol. PP, No. 99, 2016.

[11] Z. Shelby: ”CoRE Resource Directory”, draft-ietf-core-resource-directory-05, 2015.

[12] W. Colitti, K. Steenhaut, N. De Caro, B. Buta and V. Dobrota: ”Evaluation of Constrained Application Protocol for Wireless Sensor Networks”, IEEE Workshop on Local & Metropolitan Area Networks (LANMAN), 2011.

[13] Oscar Garcia-Morchon, Sye-Loong Keoh, Sandeep S. Kumar, Pedro Moreno-Sanchez, Francisco Vidal-Meca and Jan Henrik Ziegeldorf: ”Securing the IP-based internet of things with HIP and DTLS”, ACM Conference on Security and privacy in wireless and mobile networks (WiSec), 2013.

[14] D. Hardt: The OAuth 2.0 Authorization Framework, RFC 6749, 2012

[15] Girum Ketema Teklemariam, Jeroen Hoebeke, Ingrid Moerman and Piet Demeester: ”Facilitating the Creation of IoT Applications Through Conditional Observations in CoAP”, EURASIP Journal on Wireless Communications and Networking, 2013.

[16] D. Sacramento, G. Schütz and N. Correia: ”Aggregation and Scheduling in CoAP/Observe Based Wireless Sensor Networks”, IEEE International Conference on Communications (ICC), June 2015.

[17] Furuzan Atay Onat, Ivan Stojmenovic: ”Generating Random Graphs for Wireless Actuator Networks”, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2007.

[18] J. Coimbra, G. Schütz and N. Correia: ”Network Game Based Routing for Energy Efficient Fibre-Wireless Access Networks”, IEEE International Conference on Communications (ICC), June 2012.

[19] Ying Jian, Shigang Chen, Shiping Chen and Yibei Ling: ”Fair End-to-End Bandwidth Distribution in Wireless Sensor Networks”, IEEE International Conference on Communications (ICC) 2010.

[20] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler: ”Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, RFC 4944, 2007.