LRD2: *Low Resource Device Description for Energy Efficient Device Discovery*

Juan Pablo Suarez
Grenoble University
Grenoble, France
Juan-Pablo.Suarez-Coloma@imag.fr

Levent Gürgen
French Alternative Energies and Atomic Energy Commission
CEA-LETI Minatec Campus
Grenoble, France
levent.gurgen@cea.fr

Abstract—**The success of the large scale deployment of sensor actuator devices lies in their plug&play capabilities: they should be automatically discovered and ready to be used when they join to an environment. Self-description and discovery gain therefore a particular importance. Nevertheless, there is currently no largely adopted energy efficient device description and discovery standard. Existing protocols use proprietary device description models and discovery mechanisms that are incompatible between the two. This paper presents LRD2 (Low Resource Device Description), a generic description model capable of describing different kinds of device information. LRD2 implements a compression algorithm to reduce the size of description documents, thus saving energy by reducing the number of messages sent in the network. Experimental results about the performance of LRD2 are also presented.**

Keywords - WSN; Device description; plug&play; low power; energy conservation; compression algorithm

I. INTRODUCTION

Wireless sensor and actuator networks (WSAN) are used increasingly by numerous applications from various domains such as home, industrial, environmental and medical. They bring us one step closer to the “internet of things” paradigm. However, WSAN are still application-specific networks, contrary to the “general purpose” nature of the current internet. In fact, many standardization groups or industrial private solutions define stacks of protocols, targeting a particular set of domains, which are in general incompatible with the ones defined by others. This heterogeneity is the main obstacle for dynamic plug&play WSAN that is essential for successful large scale deployment of cross-domain solutions [1]. In fact in such dynamic large scale networks, devices should be self-described and self-discovered in a dynamic manner, with minimum human intervention.

Some recent efforts aim at taking on a layer-based approach, thus defining standards at different OSI layers; e.g., IEEE 802.15.4 [2] at the link layer, IETF RPL [3] at the network layer, or IETF COAP [4] at the application layer. This is an important progress towards a plug&play interoperable WSAN solution. However, at the application layer we still need to define what the devices are capable of doing and how to interact with them.

In this paper we propose LRD2, a device description model that aims to fill this gap. The main goal is to obtain a generic device description model for self-discovery of device capabilities, while taking into account energy constraints of the tiny sensor/actuator devices. The description model is based on an extensible hierarchical structure. It also implements a compression (and decompression) process. Besides being generic and extensible, the model therefore aims at being lightweight to keep the description size as small as possible. We apply a compression algorithm to the description in order to reduce its size, thus consuming less energy during its transfer to other devices in the network.

We have performed experiments to measure the efficiency of LRD2 in terms of code size, number of exchanged discovery messages, and the energy consumption for the discovery. We also compared these results against another compression mechanism, namely EXI (Efficient XML Interchange), which is a compact representation model for XML documents [5].

The paper is organized as follows: Section II presents the related work in the domain, in particular making a synthesis of related existing standards. Section III presents our proposition, LRD2, with its description model and compression algorithm. Section IV provides the results of some experiments that we have performed with LRD2. Finally, Section V concludes the paper.

II. RELATED WORK

Several WSAN standards have been defined targeting different OSI levels (from physical to application). For instance, industrial alliances such as LonWorks [6], EnOcean [7], Z-Wave [8] and Insteon [9] build solutions with a holistic approach covering requirements from the physical to the application layer. Some other standards focus on only a few layers. For example IEEE 802.15.4 [2] is a PHY/MAC layer protocol targeting low resource devices. Zigbee alliance [10] defines a set of protocols from the network to the application layer and is based on the 802.15.4 protocol. Similarly, WirelessHART [11] defines a protocol stack from network to application layer, which is also based on the 802.15.4 protocol.

	LonWorks	EnOcean	Z-Wave	Insteon	Wireless HART	ZigBee	IEEE 802.15.4	6LoWPAN	UPnP	DPWS	SensorML
Application	■	■	■	■	■	■			■	■	■
Network/ Transport	■	■	■	■	■	■		■			
Link/ Physique	■	■	■	■			■				

TABLE I. WWSAN PROTOCOLS AND OSI LAYERS THEY DEAL WITH

At the network layer, IETF specifies 6LoWPAN [12] bringing the IPv6 features to low power personal area networks on top of the 802.15.4 protocol. It is likely to be the "standard" at the network layer for personal area networks. For instance, the Smart Energy v2 of Zigbee will be agnostic to MAC/PHY protocol and will be based on 6LoWPAN.

UPnP [13] is an application layer protocol defining how to interact home appliances, in particular media devices. Similarly, DPWS (Device Profile for Web Services) [14] brings the service-oriented approach over the resource-constrained devices. UPnP and DPWS uses XML based device descriptors that are self-descriptive and expressive, yet greedy for sensor devices. SensorML [15] aims to describe powerful sensor devices rather than low power tiny wireless sensor devices.

The protocols dealing with the application layer adopt different approaches to describe devices. We can classify them into two groups according to the level of the structure of the data: **structured and semi-structured**.

The well-structured format uses fewer resources to express the information, but requires that the description receiver knows the strict structure format. On the other hand, in the case of the semi-structured format, a high level schema is enough to use the description information. However, devices need more resources to express the same information in the latter case. Protocols such as Insteon, EnOcean, Zigbee and LonWorks use a highly structured format for device description, with a few bits to express the sensor device identity and other information small in size. WirelessHART, UPnP, DPWS and SensorML provide a flexible description using a semi-structured format, mostly based on XML. The description size for these protocols is quite high to be stored and shared by very low-power sensor devices. Figure 1 shows the relation between the device description size versus the flexibility for different protocols and the position where LRD2 aims to achieve.

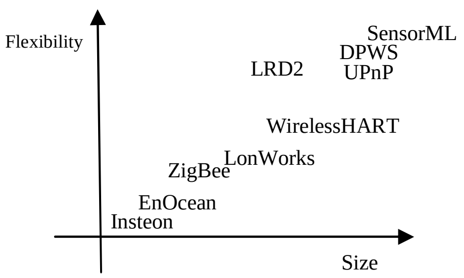

Figure 1. Description size versus model flexibility

The goal of a recent effort from the W3C, namely EXI (Efficient XML Interchange) [5], is to reduce the size of XML documents by compressing them. The idea is to assign a small binary code to the most-used elements tags, and a larger one for the less-used (instead of assigning the same quantity of bits to represent each different tag) in order to achieve a higher compactness.

The objective of LRD2 is to provide an extensible yet small in size description model. It is based on a hierarchical structure like XML that gives certain flexibility, while being less verbose in order to reach the same size of the structured data models. Similarly to EXI, LRD2 proposes a compression process for the device description with the goal of reducing the description size in the description exchange phase, therefore reducing the number of exchanged messages and saving energy consumption at devices.

III. LRD2

This section describes LRD2 device model, its compression mechanism and theory of operation of a discovery mechanism.

A. *Device description*

LRD2 is a flexible and hierarchical description model capable of describing different kinds of device information. The model schema is composed of the following elements:

- The **group** (G) tag defines a set of information under a common group, e.g., application specific information, network parameters, system information.
- The **resource** (R) tag represents the values of the resources in a group, e.g., temperature value, network address, OS name.
- The **detail** (D) tag allows the model to give some more detail on the resource information in terms of attributes, e.g., the minimum temperature measurable by a temperature sensor, network addressing type, OS version.
- The **operation** (O) tag is used to describe the specific operations implemented by the sensor device, such as sleep, reboot or ping. It defines a **parameter** (P) attribute is used to determine the input parameters of operations.
- The **enumValue** (E) tag is used to define possible values for resources or parameters.

All the tags are accompanied by a specifically chosen **qname** (qualified name) and an **ID** that allows the interaction between devices. The model also defines a *size-bits* attribute which gives the size used for the *resource* or *parameter* of operations.

LRD2 attempts to obtain small-sized yet extensible descriptions. While being strict within a group, it allows flexibility by giving the possibility of adding groups. Within groups the order of appearance of the information in the description has to be strictly respected using the specified number of bits (8 by default) in the *size_bits* tag attribute. Respecting the order allows us to formulate highly structured messages with a small size. Figure 2 gives a description example.

```
<G>SENSOR 0
  <G> APPLICATION_INFO 1
      <R SIZE_BITS=53>TEMPERATURE 2
          <D SIZE_BITS=2> ID 7 <\D>
          <D SIZE_BITS=16> MIN_VALUE 8 <\D>
          <D SIZE_BITS=16> MAX_VALUE 9 <\D>
          <D SIZE_BITS=16> MEASURED_VALUE 10 <\D>
           <D> UNIT 11
               <E>CELSIUS 1 <\E>
               <E>FAHRENHEIT 2 <\E>
           <\D>
       <\R>
       <G>SPECIFIC_METHODS 3
           <O RETURN=FALSE>SLEEP 4
               <P SIZE_BITS=10> TIME <\P>
                   <P>UNITS
                        <E>SECONDS 1 <\E>
                        <E>MILLISECONDS 2 <\E>
                   <\P>
           <\O>
        <\G>
    <\G>
<\G>
```

Figure 2. A simple description example

B. Description compressing/decompressing

Once the description is complete, an ID is designed for each word used. To achieve a higher compactness, the words used in the description are uppercase represented in 6 bits. The compressed description is composed of three parts:

The first part is the new character codification; instead of using 16 bits as UTF-8 or 8 bits as ASCII, we choose the minimal amount needed to code all the used characters in the description. In order for this message to be understandable by other devices, the first part is built as follows: 7 bits representing a digit *k* in ASCII, *k* being the minimal number of bits used for the chosen character codification, for each used character, the 7-bits ASCII representation followed by the new character codification in *k* bits. If *k* is 5, that means that the used characters are less than 64, if A and B are used characters coded as 00001 and 00010 respectively, the first codification message could start as 0110101 1000001 00001 1000010 00010 (followed by all the other coded characters), being the underlining bits an ASCII representation.

The second part of the compressed description is the dictionary. The dictionary is a stream of bits. For separating two different words (the words have a variable size) the character ":" is used. The three first words in the dictionary message are: *n1*, *n2* and *n3*; *n1* being the number of bits assigned to code the ID used in the interaction messages to access the *group*, *resource*, *detail* or *operation*; *n2* is the number of bits used to represent the size of the biggest enumValue group; and *n3* is the number of bits used to code each word in the description. After these 3 first words, each other word in the dictionary is written followed by its respective ID in hexadecimal mode and separated also by the character ":". All the characters used in the second compression part are written using the selected character codification of the first compression part.

For the third and last part of the compression process, the complete description is written using *n3* bits with each word representing the word ID assigned in the second compression part. The compression process is done only once (or each time that the description changes) directly in the description owner device. Another approach is to make the description compression and store it compressed in development time into the device, thus saving space. The compressed description is shared with the interested devices and decompressed only when it arrives at its destination.

When a device joins a network, or when it is contacted by another device, a presentation message is sent containing the three parts of the compressed description. Using these three parts, the description can be decompressed and rebuilt. The information supplied in the description is enough to interact with the sensor device containing sensor parameters and the list of operations that can be executed on the device, besides simple get/set operations to retrieve/modify parameter values. For the construction of the interaction messages, the *n1* bits ID is used to specify the information to be affected (group, resource, detail, operation). The *n2* bits are also used in the case where *enumValue* is specified. The answer interaction messages come with the *n1* bits ID to match with the interaction message.

IV. EXPERIMENTATION

For the validation of the LRD2 approach we have constructed device descriptions of different sizes and measured various values such as compressed document size, compression duration, and energy consumed on sensors to perform the compression.

We used EXI as a reference and compared the measured values with the ones obtained by EXI. We used the EXIficient v0.5 [16] implementation of the EXI. We conducted the tests over a desktop PC computer with 3 GB of RAM and processor of 2.66 GHz. Figure 3 shows the size of documents after compression by LRD2, EXI Schema-less and EXI Schema-Informed. In fact, EXI can function in 2 different modes: i) schema-less where there is no knowledge on the document's XML schema; ii) schema-informed where the schema of the document is used to optimize the compression. In the latter case, the schema is compressed using EXI schema-less algorithm, and then the schema is decompressed and used for decompressing the schema-informed compressed description.

We observed compression ratios varying from 0.71 to 0.26 depending on the size of the documents we used. The ratios we obtained are very close to the ones of the EXI's schema-less compressed documents. EXI Schema-Informed documents have greater size at the beginning as they also contain the schema of the document.

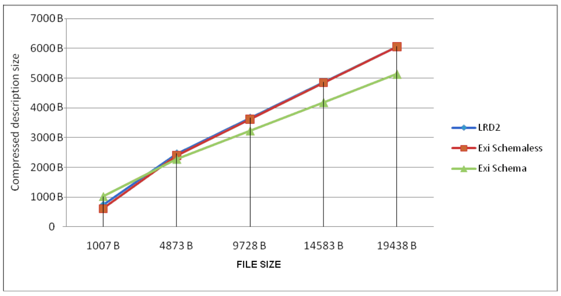

Figure 3. Compressed documents size

We also measured the description compression times with both LRD2 and EXI. Figure 4 shows the values we obtained. Even if the execution time for LRD2 compression is slightly longer than that of EXI, considering the code size of the two solutions (50 KB for LRD2 and 1500KB for EXI), LRD2 is preferable for resource-constrained devices having a small memory capacity. LRD2 is simple and lightweight and worked seamlessly over all the tested Java environments (Standard and Micro Editions).

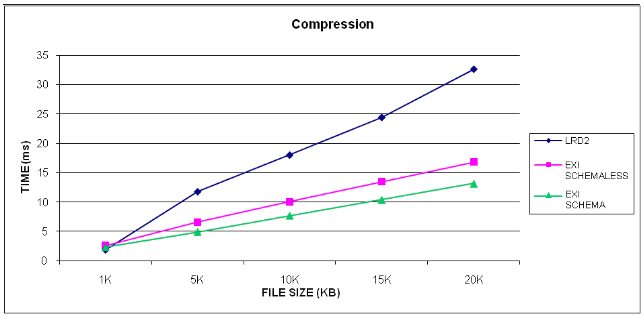

Figure 4. Compression time (milliseconds)

For the battery consumption experimentation on sensors, we used SunSPOT sensors with the following processor board specifications: 180 MHz 32 bit ARM920T core - 512K RAM/4M Flash; 2.4 GHz IEEE 802.15.4 radio with integrated antenna; 3.7V rechargeable 720 mAh lithium-ion battery. We measured on a SunSPOT node the battery capacity difference between before the start of the compression process and after the transmission of the compressed description. Figure 5 shows the measurements we obtained. We observe that energy consumption is linearly increasing w.r.t the description size and consumes between 0.04 % and 0.5 % of overall battery for a description size of between 1KB and 20KB.

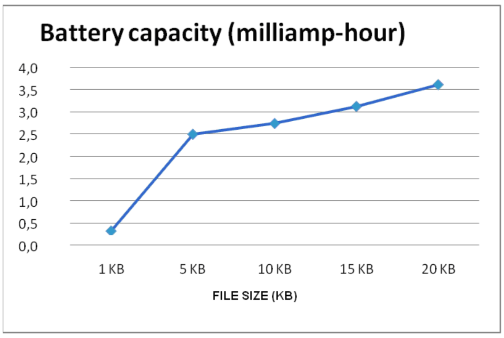

Figure 5. Battery consumption of compression (milliampere-hour)

V. Conclusion and Future Works

For successful large-scale deployment of WSAN applications, sensor and actuator devices need to be automatically discovered and ready to be used once they are discovered. Device descriptions have an important role in defining generic yet extensible descriptions to take into account new devices and technologies appearing every day in the market, as well as their constraints in terms of resources.

We have proposed a simple device description model and a compression mechanism. The semi-structural and hierarchical model lets us achieve a tradeoff between the flexibility and the size of descriptions. Thanks to its compression mechanism, for descriptions ranging in size from 1KB to 20KB we were able to obtain compression ratios from 0.71 to 0.26; compression times from 1.86ms to 32.6ms; and energy consumption values from 0.32mAh to 3.61mAh. These are reasonable values considering the fact that the compression is performed (mostly) only once. The values are close to the ones we obtained with the EXI implementation; while furthermore the code size of LRD2 being 3.3% of the one of EXI.

Our next plan is to further evaluate the energy consumption, not only at the device level but also at the whole network level. In such multi-hop networks, the energy consumed at a node to compress a description may be largely dominated by the potential energy gain in the multi-hop network, thanks to the decreasing number of exchanged messages.

Acknowledgment

This work has been partially supported by the European project OUTSMART FP7-2011-ICT-FI- 285038.

References

[1] L.Gürgen, J.Nyström-Persson, A.Cherbal, C.Labbé, C.Roncancio, and S.Honiden. Plug&manage heterogenuous sensing devices. In Proceedings of the 6th International Workshop on Data Management for Sensor Networks (DMSN'09), in conjunction with VLDB'09. 2009.

[2] IEEE P802.15^{TM} Working Group, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Standard, March 2009

[3] RPL: IPv6 Routing Protocol for Low power and Lossy Networks, draft-ietf-roll-rpl-19 (work in progress), March 2011

[4] Constrained Application Protocol (CoAP), draft-ietf-core-coap-05 (work in progress), March 2011

[5] Efficient XML Interchange (EXI) Format 1.0, W3C Recommendation, March 2011

[6] LonMark. Device Interface File Reference Guide, Revision 4.402, 2009

[7] EnOcean Alliance – Technical Task Group Interoperability, "EnOcean Equipment Profiles (EEP)", Version 2.1, January 2011

[8] Z-wave Protocol Overview, Version 4, May 2007

[9] P. Darbee, Insteon Command Tables, revision 20070927a, 2007.

[10] ZigBee Specifications, http://www.zigbee.org/

[11] WirelessHart , "Wireless Command Specification", September 2007

[12] IPv6 over Low Power WPAN (6LoWPAN). November 2009

[13] UPnP Device Architecture, Version 1.1, October 2008

[14] Devices Profile for Web Services (DPWS), Version 1.1, July 2009

[15] OpenGIS Sensor Model Language (SensorML), Implementation Specification", July 2007

[16] Open source implementation of W3C Efficient XML Interchange (EXI). http://exificient.sourceforge.net