DASH7 Alliance Protocol in Monitoring Applications

CONFERENCE PAPER · NOVEMBER 2015

DOI: 10.1109/3PGCIC.2015.93

7 AUTHORS, INCLUDING:

Glenn Ergeerts

University of Antwerp

Tomasz Surmacz

Wroclaw University of Technology

Paul De Meulenaere

University of Antwerp

Maarten Weyn

University of Antwerp / iMinds

DASH7 Alliance Protocol in Monitoring Applications

Glenn Ergeerts†, Maciej Nikodem‡, Dragan Subotic†, Tomasz Surmacz‡, Bartosz Wojciechowski‡,
Paul De Meulenaere*, Maarten Weyn†
†MOSAIC, Faculty of Applied Engineering, University of Antwerp - iMinds,
Paardenmarkt 92, 2000 Antwerp, Belgium
*CoSys-Lab, Faculty of Applied Engineering, University of Antwerp
‡Department of Computer Engineering, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
glenn.ergeerts@uantwerpen.be, bartosz.wojciechowski@pwr.edu.pl

***Abstract*—In this paper we introduce important aspects of the recently published DASH7 Alliance Protocol v1.0 specification for wireless sensor and actuator networks. The main contribution of this paper is the discussion of the different communication schemes and the accompanying trade-offs which can be used when designing a DASH7 network. Finally, we describe two practical use cases as examples of how DASH7 can be used to efficiently solve specific problems as well as the hardware developed that uses energy harvesting.**

I. Introduction

Wireless Sensor Networks (WSNs) are emerging as a viable solution for many industrial and consumer applications where traditionally wired sensors are used. They consist of small devices called *nodes* that perform measurements and communicate their results over the wireless medium. Due to their nature they can be deployed at a smaller cost compared to wired monitoring installations. WSNs are well suited for a plethora of tasks: environment monitoring, goods and equipment tracking, presence sensing and so on. Since the introduction of WSNs, their scale, number of connected nodes and possible applications expanded to the level that once was unimaginable – the connection of devices, machines, personal devices (things), cloud storage and cloud computing (Internet) created what is now called Internet of Things (IoT). Commonly, WSN/IoT nodes are battery-operated. This provides the relative ease of deployment, however it adds strict constraints on allowable power consumption of these devices. Low power consumption is achieved by reducing wireless transmission power and maximizing the percentage of time the nodes are in low-power mode. Since WSN/IoT are often deployed in relatively large environments, with potentially harsh conditions for RF propagation, transmission range is another major concern. Using multi-hop mesh protocols, such as ZigBee [1] or WirelessHART [2], or increased signal strength can be a solution, however both have a clear impact on energy consumption. Compared to IEEE 802.15.4 [3] networks D7AP is mainly focused on low-power, medium range and single hop communication. In contrast to IEEE 802.15.4 solutions it is not aiming to achieve high data rates – thus it may operate in lower frequencies, use narrow frequency bands as well as more power efficient radio modulations, coding schemes and operation modes that impose lower communication, storage and computational overhead. Simultaneously, use of lower frequencies allows to extend effective communication range at the same energy cost thus extending set of possible applications.

In this paper we show that using sub-GHz bands and a star topology can provide an energy efficient solution for medium range WSN/IoT use cases.

II. DASH7 Alliance Protocol 1.0

The DASH7 Alliance Protocol (D7AP) [4] is an open wireless sensor and actuator network standard, maintained by the DASH7 Alliance. Historically, D7AP originates from ISO/IEC 18000-7 [5]. The original standard was extended to support WSN/IoT functionalities like tag-talks-first communication [6] instead of only focusing on RFID applications. While ISO/IEC 18000-7 defined an air interface only for the 433 MHz band this is extended in D7AP to all sub-GHz ISM and SRD bands (433, 868 and 915 MHz). In April 2015, the Alliance published version 1.0 of the specification. It is out of scope for this paper to provide a complete overview of the specification, instead we refer to the specification [4] itself and to a previous paper describing D7AP v0.2 [7]. Nevertheless, we will highlight a few important concepts of the specification, like the design rationale, the possible communication schemes and physical layer properties, which are necessary to understand how a DASH7 network can be configured to provide a solution for different use cases.

D7AP is conceived as a medium range, asynchronous network protocol enabling communicating small and bursty data packets between devices in a request-response fashion. Endpoints are considered to be energy constrained and mobile within the environment. Moreover, the whole network of devices is assumed to be transitional and there is no need to maintain routing information or other management information. Larger communication range allows to use a star or optionally a tree topology using maximum one hop between the endpoint and a gateway as opposed to a more complex mesh networking topology. Endpoints can however enforce that the packet is acknowledged by at least one gateway, thus providing sensor-to-cloud communication. D7AP covers all OSI-layers, from the physical layer up to the application layer. Central to the D7AP stack is the concept of a file system. The D7AP file system is a collection of structured Data Elements together with their associated properties like permissions and storage class. Data Elements are defined for D7AP configuration parameters used by the stack, like the encryption key or the Access Profiles (which may describe the channel, coding and TX power). Next to the D7AP System Files there is a range of Data Elements which are reserved for application usage, to store sensor readings for example. The application manages the Data Elements through the D7AP Application Layer Programming Interface (ALP). An ALP Command consists of one or more ALP Actions and each Action specifies an operation to be executed – e.g. read, write, or execute file, etc. A set of Actions can be preceded with Queries, where the Boolean output of the Queries determines if the Actions should be processed or not. A query is defined as an arithmetic comparison or string token search between data in a file and supplied data, or data in another file. Queries in ALP Commands are used as a flexible addressing mechanism. A gateway or subcontroller uses ALP Commands to interrogate and manage the endpoint nodes in the environment. For instance, using an ALP Command containing a Query and a read file Action we can instruct all nodes which detect a temperature higher than 50 °C to respond. More complex combinations using multiple Query operands, chained together with boolean operators, and multiple Actions are possible as well. A network of DASH7 nodes effectively behaves as a distributed database, where the query is processed on the endpoint nodes transmitting the requested data only if needed.

To maximize global availability the D7AP PHY supports the 433, 868 and 915 MHz unlicensed ISM/SRD bands. The specification defines three channel classes: Lo-Rate, Normal and Hi-Rate, using a symbol rate of 9.6, 55.555 and 166.667 kbps respectively. All channel classes are modulated with GFSK but using different frequency deviations and channel spacings. The number of channels available varies per band and per channel type on one hand, and local regulatory limitations which might disallow usage of parts of the band on the other hand. Besides PN9 data whitening the specification also allows 1/2-rate convolutional Forward Error Correction (FEC) channel coding. The combination of the frequency band, data rate, transmit power and coding parameters together with the constraints like range, antenna size, cost and energy consumption results in a trade-off which should be made for every application. For instance, using the 868 MHz band instead of the 433 MHz band allows decreasing antenna size. In another case where you want to maximize the range and you have enough power budget available you can opt for 433 MHz band using a Lo-Rate channel and FEC encoding.

III. D7AP Communication Schemes

In this section we introduce different communication schemes a system architect can use, or combine, when designing a solution using D7AP.

A. Polling data using the D7AP Advertising Protocol

A gateway or a sub-controller can query endpoint nodes by transmitting ALP Commands. An endpoint is typically constrained in terms of energy consumption however, thus it is not expected to continuously scan the channel for requests. Furthermore, DASH7 is designed to be an asynchronous protocol, to prevent the periodic network synchronization overhead. Instead, D7AP uses a low-power wake-up mechanism, called the D7AP Advertising Protocol (D7AAdvP), which is effectively an ad-hoc synchronization performed only when a request is queued. The D7AAdvP is started by the transmitter (gateway or sub-controller) which floods the channel by continuously sending background advertising frames. These frames contain the Estimated Time of Arrival (ETA) of the foreground frame containing the actual ALP command. The subsequent advertisement frames thus are counting down until the ETA reaches 0. Receivers, on the other hand, are configured to schedule a background scan at a certain rate. If no modulated signal is detected the endpoint returns to low-power mode immediately until the scheduler interval elapses again. When a signal is detected the endpoint starts scanning for a background advertising frame. When this is successfully received (within a timeout) the endpoint knows the ETA of the pending foreground request and returns to low-power mode until this period has elapsed after which the endpoint will switch to foreground scanning mode to receive and subsequently process the request. This scheme is ideal for cases where you need to poll the node's sensor data in an ad-hoc fashion, typically not using a fixed periodicity or not triggered by sensor data events but rather triggered by some business-logic running on the infrastructure (e.g. the gateway or a cloud server). For instance, an end-user who interacts with a user interface to request live data from the network might initiate such a polling sequence. One important design consideration to make in this scheme is the trade-off between latency and energy consumption. Increasing the rate at which endpoints scan for background frames decreases the total time needed for fetching the data but will have a negative impact on battery live of the endpoints, and vice versa. It should be noted, however, that these parameters are all contained in files as well, and thus are updatable over the air (given the correct authentication), allowing for post roll-out tuning of the network.

B. Pushing data using the D7AP Action Protocol

The D7AP Action Protocol (D7AActP) provides a way to use ALP Commands to push data using a tag-talks-first scheme instead of the querying scheme as explained above. The same ALP Command which would be used for querying is preregistered on the endpoints in a file. Normal data files can be configured to use D7AActP which will trigger the ALP Command contained in the file pointed to upon reading or writing the file. An endpoint application only needs to write sensor data periodically to a file. Based on the file properties and the ALP Command in the action file (if enabled), the

D7AP stack will take care of processing the actions (conditionally if the Command contains a Query) and transmitting the request. While this may be achieved by polling the network using the same ALP Command as well, the main advantage of this scheme is that for many cases it is more efficient to push the data. For instance, if we want to be sure we always receive the node's sensor data (optionally validating a query condition) we would need to poll using at least the sensor update rate. By using the D7AActP we do not need to poll anymore to receive the same data. Clearly, this is more efficient in terms of energy consumption and spectrum usage for cases that are based on triggering upon sensor data changes on the node itself. Again, like most D7AP configuration parameters, the file action properties and the preregistered ALP Commands on the nodes can be adapted over the air interface.

C. Dormant Sessions

In cases where data is being pushed from the endpoint to the infrastructure (using D7AActP as described above), it might be necessary to occasionally send data to the endpoint as well, e.g. to change configuration or drive an actuator. This is achievable by using the D7AAdvP to notify endpoints of a pending request, as described above. In this particular case however, the concept of Dormant Sessions can be employed to avoid using the relatively expensive D7AAdvP mechanism. When a gateway has a pending request for an endpoint it can be queued in a Dormant Session together with an associated timeout, instead of directly transmitting the request. When the timeout is reached the Dormant Session is made active and the request will be transmitted, using D7AAdvP synchronization. However, if the addressed endpoint happens to start a dialog with the gateway before the timeout has elapsed, the gateway can return a flag in its response to the endpoint, signalling a pending request. From this moment on the dialog is extended and the gateway becomes the requester and the endpoint the responder. Next, the request is transmitted as part of this extended dialog. If the use case allows a downlink latency equal to the (maximum) time between two uplink push message then this method provides a much more efficient way compared to using D7AActP, since the endpoints only need to switch to receive mode when they are informed there is a pending request. The D7AAdvP ad-hoc synchronization is not needed since gateway and endpoint are effectively already synchronized when participating in a dialog.

IV. Examples of Use

In the next two subsections we will describe two distinct cases and show how D7AP can be used. The final two subsections describe our development kits.

A. Bird Tracking

The goal of the bird localization system is to get an idea how the birds make their choice where they are going to settle and the behavior of the group. To do this we need to track a young bird from the beginning of his life to when it builds a new nest with its partner. The birds that will be tracked are tomtits.

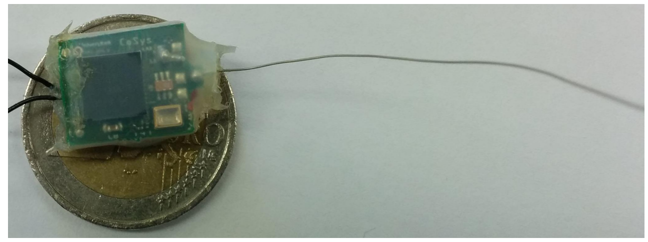

Figure 1. Endpoint developed for bird tracking project

This species is small, having a body weight of 20 gram. The allowed weight of the tag is limited to 5% of the body weight. Currently the most used bird tracking system is a leg ring based on passive RFID combined with readers placed in bird cabinets. The advantage of this setup is the low weight of the passive tag, due to the lack of a battery. The main disadvantage though is that the readers consume a lot of energy and and require the batteries to be replaced weekly, making this a very expensive system to maintain for longer periods. The system we propose in this section is based on active tags using D7AP and a low-power System-On-Chip (SoC). Compared to the previous, passive RFID system, this has two main advantages. First the range will increase significantly, from only the bird cabinet (and it's direct vicinity) to more or less 60 m, thus enabling tracking of the birds not only when they are inside the cabinets. Secondly, the power usage of the readers is limited, decreasing the necessary battery replacement frequency from weekly to monthly. Designing an active tag within the weight constraints mentioned before will be the biggest challenge. Other constraints are lifespan of the endpoint and the range between the tags. The power source will affect the lifespan of the tag but is limited in capacity by the weight constraint. The range depends on the antenna design and the TX power, which in turn is limited by battery capacity as well. The minimum number of packets per day is three, this means that the endpoint status is almost continuously in low-power sleep mode. The hardware contains a CC430 SoC based on the MSP430 microcontroller and CC1101 radio chip. In standby mode it consumes 2 μA, the energy cost to send one packet is 260 μA/s. Duration of one packet is 17.7 ms, this includes start-up time of the microcontroller, two time collision avoidance and transmitting the data. In order to fulfill the constraint of a 1 year lifespan, the battery has to have the capacity of 6.62 mAh at least. A whip antenna that is made of a single wire and placed on the back of the bird is used in the interest of the weight constraint as for the bird well-being. Using this antenna setup it is feasible to use a quarter wavelength antenna for 433 MHz of 17 cm. The tag that has been developed for this project (Fig. 1) has a CR1216 battery and currently weights 2.1 g, in the future there will be few optimizations needed to achieve the 1 gram constraint.

The challenge of this network is the environment where it is located. In this project the tree topology is used. The endpoint that is positioned on the bird sends a packet to a sub-controller. The sub-controllers are placed in bird cabinets, forwarding the

Figure 2. Test set-up for a bird tracking project

packets to the gateway. The gateway is positioned at the forest frontier. Because the range between the nodes can be few hundred meters, a sub-GHz technology is demanded here. The test set-up that is shown in Fig. 2 contains six endpoints, four sub-controllers that forward the packets and one gateway that collects all the data. Distance between the sub-controllers is approximately 40 meters, the distance from the sub-controllers to the gateway is approximately 350-400 meters. Different endpoints are placed in the trees and bushes, between 20 and 75 meters from the sub-controllers. The temperature in the cabinets is monitored using the D7AActP, if the temperature in the cabinet exceeds 45 degrees the sub-controller will push a message to the gateway. Otherwise the temperature measurement will be executed but the data will not be transmitted. The interest is to track the movements of a bird, therefore the localization accuracy does not have to be few centimeters. This is why we decided to use proximity localization based system. The endpoint will periodically broadcast a packet. The sub-controllers which receive this packet will use the received signal strength indicator (RSSI) and the transmitted power parameter to calculate the net link loss between of the nodes, which is then forwarded to the gateway. This data will be used to compute the position of the bird. From the tests that have been conducted all four sub-controllers receive messages of all six endpoints. This allows for the trade-off between being able to locate more accurately or to spread out the sub-controllers to gain larger coverage.

B. Greenhouse monitoring application

D7AP is also well suited for environment monitoring applications (such as greenhouse crop growth monitoring), providing different operation modes and communication features to achieve required functionality while ensuring longevity of the network. The sensor-to-cloud concept ensures that the exact route between the sensor and gateway is not important as the network declares transmission success if at least one of the gateways receive the packet. Consequently, the network is more robust to variations of radio propagation conditions that may result from changes in environmental conditions (e.g. humidity) or large obstacles (containers, trucks or even people working at the greenhouse). Smart addressing is another important feature that allows nodes to be addressed by the sensor types or properties. This enables the requester to ask for the information of interest (e.g. air temperature) without the need for providing responder IDs or addresses. This feature, together with D7AP transactions and dialogs, allows to reduce number of requests, shortens the transmission times and thus reduces bandwidth requirements and power consumption. A greenhouse monitoring system is also a perfect example of a network where both no-hop and one-hop routing is used. Due to the steel-and-glass construction of the buildings and a number of other systems deployed (e.g. heating, water, cabling and large automatic systems) the effective communication range is limited and routing may be needed. Nevertheless, it is desirable to deploy a network requiring no retransmission, as the ability to communicate directly with the gateways minimizes energy consumption on data transmission [8]. In a sense, greenhouse monitoring is a good pilot area for verification and evaluation of small, yet complex D7AP network.

Our test application was developed for monitoring the crop growth in the greenhouse complex owned by the Wroclaw University of Environmental and Life Sciences. The complex (Fig. 3) related to research and teaching activities is composed of 8 greenhouses – 5 of which constitute the main, interconnected complex and 3 are separate. Seven greenhouses are 10 meters wide and 30 meters long and are divided into 3 sections. The last greenhouse is wider (20 m). The Internet network access is available in a nearby administration building. The maximum distance between nodes does not exceed 100 m, but due to the obstacles in the communication path (metal construction elements and growing crops that also block radio propagation) one-to-one visibility is not possible for each network node, thus the placement of the gateways is limited. The initial requirement was to deploy at least one sensor in each greenhouse. However, preliminary results showed that environment conditions may significantly vary between sections of the same greenhouse. Similar variances were noticed for the largest greenhouse. Consequently in the test application a small network of up to twenty endpoints was used. The endpoints were deployed in the areas of interest (i.e. where crop growth was ongoing) sensing various climate parameters (e.g. temperature, humidity, luminosity, as well as CO, CO_2 concentration) and providing this information to the gateways. The network included two Internet gateways that were running on RaspberryPi boards and were interfacing the WSN network to the Internet using Wi-Fi connection and a Virtual Private Network, to bypass local network complexities (e.g. masquerading) and make them accessible from the outside. The two gateways were deployed in two different locations in order to ensure the best coverage as well as to ensure reliability of the network – so that failure of a single board did not compromise the whole system.

The first network deployment in February 2012 used TelosB WSN nodes operating in 2.4 GHz band and controlled by TinyOS system. The information from nodes was collected periodically using tag-talks-first concept – measurements were collected and reported periodically every minute, but could also be initiated by high level events, e.g. measurements above/below the predefined threshold. Additionally gateways could request selected information from the network or a selected node. In parallel to environment information the nodes were monitoring their own parameters and network operation providing information on energy consumption, remaining battery power and RSSI values measured for incoming packets. This information allowed us to monitor and adjust the operation of the nodes and the network (e.g. adjust timings, transmission powers, etc.) and provide alerts to the supervisors on possible threats (e.g. power outages or greenhouse's system failures). Gateways were able to use information on network performance in order to change settings of any endpoint in the network (e.g. increase transmission power if RSSI measurements were low). The same was available for end-users connecting to the network from the Internet, over the VPN tunnel. Proper adjustment of duty-cycling allowed to extend the unattended network operation from initial 3 weeks (on 2 AA-sized batteries) to about 6-8 weeks (before first nodes run out of battery). In a network with sub-controllers further improvement was possible when rough time synchronization between nodes was introduced.

Using D7AP helps solving many problems that were identified during the initial network setup. Using a sub-GHz band allows for longer distances between nodes, effectively eliminates the need of packet retransmissions and simplifies network operation. Endpoints talk directly to the gateway whenever they need, and if a particular node needs to be polled, the D7AAdvP protocol is used to make it aware of the upcoming data query. Single hop operation (i.e. no retransmission) also makes time synchronization and duty cycling unnecessary and saves energy by maximizing time spent in deep sleep modes. Information on net link budget enables endpoints to adjust transmission power according to the propagation conditions – increasing transmission power when higher attenuation of radio signal is detected and decreasing it when communication channel improves. As assessment of the channel quality is a byproduct of a regular communication, it brings no additional cost to the system and allows for further energy savings [9]. D7AP is a transitive protocol, so nodes are not connected to any gateway. This allows using multiple gateways and packets can be transmitted to the Internet over any gateway. Gateways can be accessed over the Internet using VPN, providing online access to the measurements and enabling remote configuration of the devices.

C. Evaluation boards with energy harvesting

Although use of D7AP allows to significantly reduce power requirements of individual nodes and the network, the energy available in a typical WSN/IoT node is limited by the capacity of the battery and limits the network operation time. Further extension of network life-time is possible if the node can harvest energy and recharge batteries. In such case it might be even possible for the endpoint to operate infinitely long.

At Wrocław University of Technology we have developed a ForEverSense (FES) node – an MSP430-based WSN/IoT node that can be used for D7AP protocol development, testing, implementation of proof-of-concept solutions and evaluation

Figure 3. Overview of the greenhouse complex with positions of network nodes. Green triangles denote location of endpoints, red circles show positions of two gateways, and the blue rectangle is the location of WiFi access point.

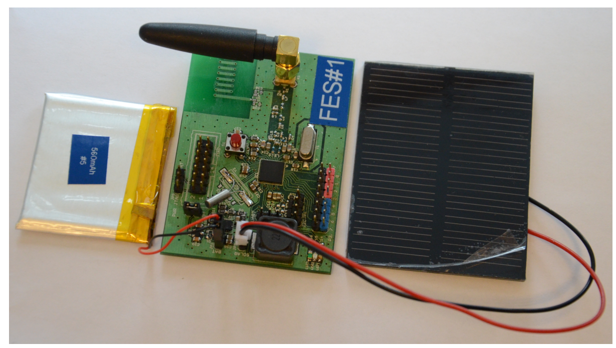

Figure 4. FES – a WSN/IoT prototype node incorporating an energy harvesting module

(Fig. 4). FES integrates two independent modules – a low-power wireless sensor, and an energy harvester. The wireless module uses CC430F5137 processor which is a SoC that contains a microcontroller and radio transceiver. The board contains also an on-board PCB antenna designed for 868 MHz band with dimensions of 12×20 mm. Alternatively, an SMA socket is provided for external antenna and antenna diversity operation.

The energy harvesting part is designed to use an inexpensive solar panel with small size that is comparable to a credit card. Solar energy harvested is stored in a LiPol battery (small batteries between 560 and 950 mAh were used). For a typical wireless sensor nodes, which requires below 0.5 mA in sleep modes and below 15 mA while transmitting or idle-listening, this provides enough power to sustain sensor operation. This is also true for long periods of bad weather conditions when less energy is harvested. From the experiments conducted during winter, when the sun exposure is the shortest and lighting conditions are the worst, at most 3 sunny days were enough to compensate for battery power loss caused by 2 weeks of operation in cloudy weather with no direct sunlight (Fig. 5).

As mentioned, the FES board is designed for easy protocol development, so most of the processor outputs are available

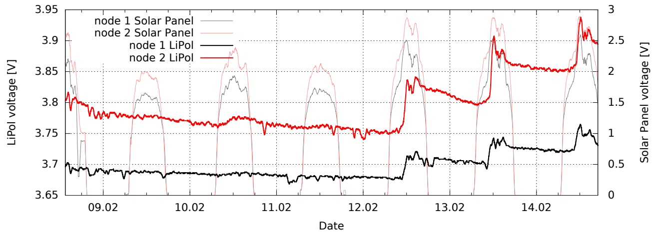

Figure 5. Battery charging/discharging of two FES nodes and corresponding solar panel voltage

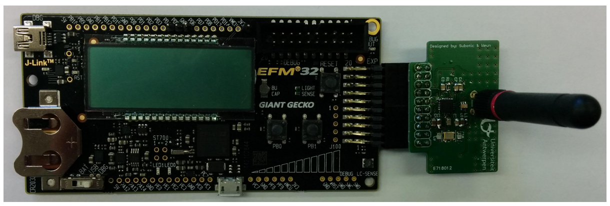

Figure 6. Development kit based on STK3700 and CC1101

at board connectors, which allows using additional sensors, as well as communicating over RS232 with a host computer. This opportunity was used to construct a gateway node that was based on Raspberry Pi.

D. Development kit

At MOSAIC we are using a Silicon Labs EFM32GG-STK3700 as a development platform. This board is based on an ARM Cortex M3 MCU and contains several sensors, buttons, LEDs, pin headers and an LCD for rapid prototyping. We designed a RF module based on TI CC1101 chip which fits in the expansion port of the STK3700 board. Antennas can be connected using an SMA connector, giving the possibility for easy performance testing of antennas. The OSS-7 open source stack [7] currently has the best support for this platform.

V. Conclusions and Future Work

In this paper we introduced D7AP v1.0 and showed how the specification can be used to provide solutions to mid-range WSN/IoT use cases in an original and efficient way. The versatility of the different communication schemes and the accompanying trade-offs were explained as well as the ability to reconfigure the behavior of nodes deployed in the field using standard file operations, without performing firmware updates. Next we described two different use cases where D7AP and the better propagation properties of sub-GHz communication are a good match.

D7AP has numerous possible applications in mid-range, low-power WSN/IoT networks. Extended communication range, compared to IEEE 802.15.4 networks, and low power operation broadens the area of possible applications. For example, the use of D7AP improves operation of localization and tracking systems based on RSSI measurements. This is achieved thanks to the fact that, for the same deployment, the average number of reference nodes (sub-controllers or gateways) in the communication range of a single endpoint increases. Consequently, the location of every mobile endpoint is determined with respect to a larger number of reference nodes which in turn allows for improved localization. Moreover, inherent transitional operation and lack of routing information maintained by nodes of the network simplifies tracking and "handovers" in large networks that extend beyond the communication range of a single gateway. Larger link budgets and better penetration of sub-GHz waves through obstacles make D7AP also more suitable for indoor localization applications while still ensuring low-power operation. Therefore, RSSI-based localization algorithms and procedures for D7AP networks will gain more of our attention in future.

Regarding the FES node we plan to split its modules into separate devices that can be easily connected with other modules in order to provide different configurations suitable for vast range of applications. We will therefore separate harvesting, SoC and sensor modules – simultaneously minimizing their size and weight so that possible application range can be further extended. We plan to improve the harvesting module so it will allow to harvest energy from different sources (e.g. heat, wind, salinity or motion) and use small battery packs (as this directly influences the size and weight).

For the bird tracking case we are planning a new design of the tag where the CC430 SoC is replaced with separate Cortex-M0+ MCU and CC1101 RF chip, allowing to further shrink both the footprint and the weight of the tag. Furthermore we are planning to use the lo-rate mode which lowers the datarate from 55.555 kbps to 9.6 kbps which should increase the range. Finally, we will start testing the system on actual birds instead of only using fixed reference tags.

References

[1] S. Farahani, *ZigBee Wireless Networks and Transceivers*. Newton, MA, USA: Newnes, 2008.

[2] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in realtime industrial process control," in *Real-Time and Embedded Technology and Applications Symposium, IEEE*, 2008, pp. 377–386.

[3] IEEE Computer Society, "IEEE 802.15.4 – IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2011.

[4] *DASH7 Alliance Wireless Sensor and Actuator Protocol*, DASH7 Alliance Std., Rev. v1.0, April 2015.

[5] *ISO/IEC 18000-7, Information technology–Radio Frequency Identification for Item Management, Part 7: Parameters for Active Air Interface Communications at 433 MHz*, ISO Std., 2006.

[6] M. Wang and K. Yeung, "RFID Multi-hop Relay Algorithms with Active Relay Tags in Tag-Talks-First Mode," in *Computing and Networking (CANDAR), 2013 First International Symposium on*, Dec 2013, pp. 617–621.

[7] M. Weyn, G. Ergeerts, L. Wante, C. Vercauteren, and P. Hellinckx, "Survey of the DASH7 alliance protocol for 433 MHz wireless sensor communication," *International Journal of Distributed Sensor Networks*, 2013.

[8] M. Nikodem and B. Wojciechowski, "Upper Bounds on Network Lifetime for Clustered Wireless Sensor Networks," in *New Technologies, Mobility and Security (NTMS), 4th IFIP International Conference on*, Feb 2011.

[9] M. Nikodem, M. Slabicki, T. Surmacz, and B. Wojciechowski, "Transmission power control based on packet reception rate," in *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, March 2014.