DASH7 Alliance Protocol 1.0: Low-Power, Mid-Range Sensor and Actuator Communication

Maarten Weyn, Glenn Ergeerts, Rafael Berkvens
Department of Applied Engineering
University of Antwerp - iMinds
Belgium
{maarten.weyn,glenn.ergeerts,rafael.berkvens}@uantwerpen.be

Bartosz Wojciechowski
Department of Computer Engineering
Wroclaw University of Technology
Poland
bartosz.wojciechowski@pwr.edu.pl

Yordan Tabakov
Wizzilab
France
yordan@wizzilab.com

***Abstract*—This paper presents the DASH7 Alliance Protocol 1.0. It is an industry alliance standard for wireless sensor and actuator communication using the unlicensed sub-1 GHz bands. The paper explains its historic relation to active RFID standards ISO 18000-7 for 433 MHz communication, the basic concepts and communication paradigms of the protocol. Since the protocol is a full OSI stack specification, the paper discusses the implementation of every OSI layer.**

I. Introduction

The DASH7 Alliance Protocol (D7AP) is an industry alliance standard for wireless sensor and actuator communication over unlicensed Sub-1 GHz bands specified and promoted by the DASH7 Alliance. It has its origins in the ISO 18000-7 standard for active RFID, intended by the US Department of Defense for container inventory. D7AP inherits from ISO/IEC 18000-7 the default parameters of the active air interface communication at 433MHz, an asynchronous Media Access Control (MAC) and a presentation layer using exclusively highly structured data elements. D7AP significantly extends the latter standard and defines a complete communication stack for Wireless Sensor and Actuator Networks (WSAN) from the physical layer up to the application layer and ensures interoperability amongst different providers. It includes high-level functionality optimized for active RFID and WSAN applications. In contrast to legacy RFID systems [1], D7AP supports tag-to-tag communication. In 2013 the D7AP 0.2 was published by the DASH7 Alliance [2]. In April 2015, the DASH7 Alliance published D7AP 1.0. This paper will discuss this industry standard [3].

In Section II five basic concepts of the DASH7 Alliance Protocol are explained, while Section III explains the communication paradigms which organize the communication between nodes. And finally, in Section IV the different OSI layers which are defined by D7AP are explained.

II. D7AP 1.0 Concepts

D7AP is built around five major concepts: BLAST (Bursty, Light, Asynchronous, Stealth, and Transitional) communication, the D7AP data elements which are used to perform tasks as well as store data, the D7AP sessions, which allow ad-hoc communication, a preference for tree network architectures, and the D7AP physical communication properties. We will discuss these five concepts in the following subsections.

A. BLAST communication

BLAST is the acronym used to describe the key ideas of D7AP. *Bursty* indicates short and sporadic sequences of data that are transmitted on the medium. D7AP is particularly useful for sensor data collection and sensor/actuator configuration, but excludes streaming communication. *Light* stands for small data packets; their maximum size is 256 bytes. *Asynchronous* is a D7AP property that is achieved by ad-hoc synchronization between devices when communication is necessary. Periodic synchronization is not required. *Stealth* indicates that DASH7 devices can choose to only respond to previously configured devices. They will ignore requests from any other device. Moreover, no beaconing or advertising is required by the nodes. *Transitional* stands for the highly mobile support where nodes can seamlessly move between the coverage of different parts of the network.

B. D7AP Data Elements

D7AP is a full stack specification, implementing the complete OSI model, up to and including the presentation and application layer. The presentation layer consists of a complete file system of D7AP files which contain configurations of the protocol, user data, and can be executed as scripts. DASH7 applications can be built using such files. The files are highly structured data elements with additional file system properties such as the permissions and an indication of where and how the files must be stored — either volatile or non-volatile.

The protocol is focused on sensor and actuator networks. Devices can be addressed by their identifier, but also by their properties, *e.g.* whether the device has a temperature sensor, or whether the device can switch on a light. This addressing by property can happen by condition, *e.g.* only address devices with an estimated battery level under 20%. Sensors and actuators are defined by ISO 21451-7 [4], enabling data to be transmitted to or received from networks that use other protocol than DASH7.

C. D7AP Sessions

The concept of D7AP sessions is highly asynchronous in initiation. Tags can decide to send information to the gateway at any time, which is called Tag-Talk-First. The Quality-of-Service (QoS) allows the tag to wait for the gateway's

acknowledgment. In case of multiple gateways within range, the tag can also choose to perform an all-cast, where it waits for multiple the gateways to acknowledge, or any-cast, where it waits for a single gateway's acknowledgment. Gateways can also query the network for specific sensor or actuator information.

D. Network

A D7AP network consists of gateways and endpoints, and can optionally contain sub-controllers. A gateway is a device that is capable of permanently listening for packets. It receives data, can process it, but typically transmits it on another network. It can also transmit data to the DASH7 network from the other networks it is connected to. The gateway implements all D7AP features. A subcontroller can also implement all D7AP features but is allowed to sleep and can for example be used to relay packets while being optimized to minimize the power consumption. An endpoint is a simpler device that does not need to implement all D7AP features. It typically contains sensors or actuators and transmits or receives information about these. They are designed to operate at low energy consumption and spend most time in sleep mode. The endpoint can transmit asynchronously and wakes up periodically to listen for possible incoming data. Sub-controllers also have periodic listening cycles, but are different from endpoints in that they implement all D7AP features.

These devices are optimized for tree network architectures. A gateway is the root, endpoints are the leafs, and sub-controllers can be used to facilitate management of larger networks.

E. D7AP Communication

DASH7 has an asynchronous session initiation, an ad-hoc synchronization establishes a communication dialog between two or more devices. The asynchronous communication is further discussed in the next section.

The D7A protocol is designed for sub-GHz ISM bands at 433 MHz, 868 MHz, and 915 MHz. These bands can be used at low, normal, or high rate, which is respectively 9.6 kbps, 55.555 kbps, and 166.667 kbps. Depending on the speed and band, DASH7 can cover distances in the order of hundred meters to a few kilometers. In each band, DASH7 defines different channels. D7AP is naturally frequency-agile. Devices can be configured to scan for messages over more than one channel.

III. D7AP 1.0 Communication Paradigms

In this section the communication paradigms which are used to build the D7AP are discussed. In the next section the specification of the different OSI layers are discussed to enable these concepts.

A. File based messaging

D7AP 1.0 focuses on a file-based communication. The majority of the communication between the application and the communication stack is done using file access actions.

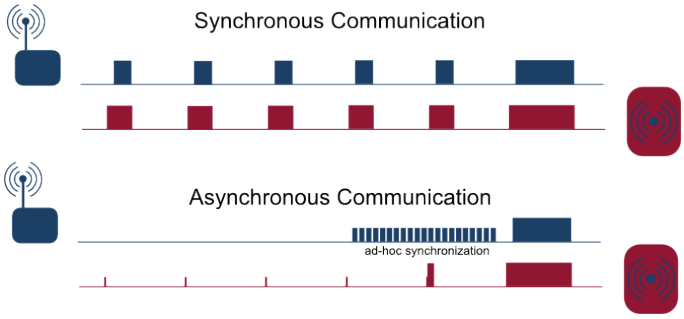

Fig. 1. D7AP 1.0 Low Power Wake-up

The API for this communication is further described in Section IV-G. The sensor or actuator data transmission using D7AP is always in the form of writing to or reading from a remote file. This makes the system very extensible.

B. Query - Response

Originating from an RFID standard, D7AP is build around a query-response communication model. Different than other protocols the addressing of the nodes is context based and not necessarily based on device addresses. This addressing also enables grouping devices into different context-based subsets in order to collect data from a larger amount of devices. The Query can trigger a direct response or can configure a device to respond on the query when a certain event happens. Responses triggered by file notifications are explained in Subsection III-D.

C. Low Power Wake-up

Sensors and actuators optimize their power consumption by limiting the active time of the node. When a node or gateway needs to query these low powers endpoints, the endpoints need to be awake. D7AP uses a low power wake-up system method to minimize the used energy. This method is shown in Figure 1. In a synchronous communication model, the endpoints needs to listen regularly to synchronization data which is sent by a gateway. In DASH7 on the other hand, the querying node (blue) sends an ad-hoc synchronization train using the D7A Advertising Protocol (D7AAdvP) as described in Subsection IV-C. The sleeping node (red) detects a signal above noise level and listens to the advertising frame which contains the time when a request will be sent. The node goes to a low power mode until time the request is expected to be received.

D. File Notification Triggers

Instead of using an over the air query, a query can be preregistered on the endpoint in a file. This can be done on the moment the endpoint is configured or later by an over the air write action of the file header. A file can be configured to use the D7A Action Protocol (D7AActP) which will trigger an ALP Command (Section IV-G), i.e. a response to a prerecorded query. Using this system an application on an endpoint only needs to write sensor data to a file, the D7AP

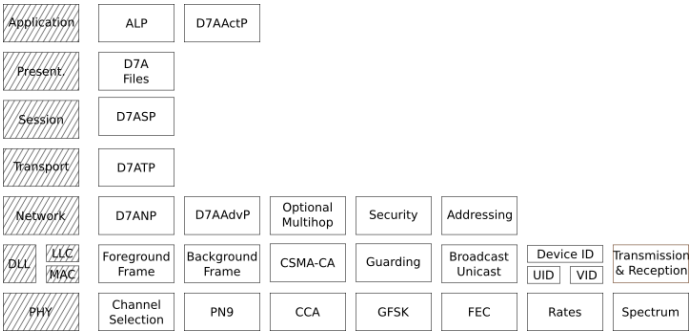

Fig. 2. D7AP 1.0 Layer Overview

TABLE I
D7AP MODULATION SCHEMES USING 2-(G)FSK

Channel Class	Channel Spacing (MHz)	Symbol Rate (kbps)	Modulation Index	Frequency Deviation (KHz)
Lo-Rate	0.025	9.6	1	± 4.8
Normal	0.200	55.555	1.8	± 50
Hi-Rate	0.200	166.667	0.5	± 41.667

stack will handle the processing of the data and transmit the data when the query which is preregistered matches the file change.

E. Dormant Sessions

As described above, a gateway can use the low power wake-up system to activate endpoints and send data. However, it can also use the concept of Dormant Sessions. When the gateway has a pending request for the endpoint, it is queued as a dormant session with a certain timeout. When this timeout expires, the session is activated and the data is sent using the D7AdvP as described above. However, when the endpoints send data to the gateway before the timeout expires, e.g. using to D7AActP, the gateway can notify the endpoint by setting a flag in the response the endpoint. From this moment the dialog is extended and the endpoints becomes the responder while the gateway becomes the requester.

IV. D7AP 1.0 LAYER OVERVIEW

In this section we will discuss some of the individual aspects which are defined in the specification. The specification is constructed based on the OSI-layers. In Figure 2 an overview is shown of these layers with the most important aspects which are defined in the different layers of the D7A protocol.

A. Physical Layer

The D7A 1.0 protocol supports three Sub-1 Ghz bands, and three data rates, as discussed above. D7AP uses 2-(G)FSK, the modulation schemes of the different channel classes are described in Table I.

In all cases PN9 encoding is used for data whitening. Optionally ½ FEC encoding can be used. In this case the first stage is the encoding using a ½ rate convolutional code with a constraint length of 4 and the second stage is a matrix interleaver to minimize the impact of burst errors. The support of both coding schemes, together with the possibility to use two different frame types (as discussed in the next session), requires that 4 different sync words are used.

TABLE II
D7AP 1.0 DATA LINK FRAME STRUCTURE

Preamble + Sync	Length	Subnet	Control	TADR	Payload	CRC
bytes:	1	1	1	0/2/8	2/8	2

B. Data Link Layer

The frame structure defined in the data link layer is defined in Table II. The target address (TADR) type is defined in the control bytes, together with the equivalent isotropically radiated power (EIRP). The TADR can be non-existing, 2 bytes in the case of a Virtual ID (VID) or 8 bytes in the case of a Unique ID (UID). The UID is a 64 bit EUI-64 value, the VID is a 16 bit ID, also compliant with ISO 15963[5].

The Data Link Layer uses filters to qualify incoming frames for processing. First CRC16 is used to validate the data, afterwards a subnet matching is done. The subnet is constructed using a specifier and a mask or identifier. Third, a link quality assessment can optionally be executed and finally a Device ID filter is used.

Moreover, D7AP defines sub-bands which enable the nodes to use a group of channels to communicate on. These sub-bands are used in a so-called access profile, which defines the subnet, a period for automatic scanning of the band, a transmission time-out and the number of the sub-bands to scan or transmit on. The automated channel scanning is controlled by the data link layer.

D7AP defines two types of frames, a background and foreground frame. The background frame is used to send short messages to control the communication, the foreground frames are used for the typical communication of data.

The automated scanning of background frames needs to detect a signal above noise level before it starts listening to the channel itself. When transmitting, a CSMA-CA system is used with 3 possible algorithms: Adaptive Increase No Division (AIND), Random Adaptive Increase No Division (RAIND) and Random Increase Geometric Division (RIGD). Channels guarding is used both during, and after the transmission of a frame.

AIND will insert the packet in the beginning of a slot, and the slot duration is approximately equal to the duration of the transmission being queued. When the transmission fails, a random wait duration is applied. RAIND is identical to AIND, but in contrast a random wait duration is applied in the initial insertion. RIGD uses a random slot insertion with a geometric decaying slot back-off.

C. Network Layer

The D7AP Network Layer defines the background and foreground network protocols. Although more background

TABLE III
D7AP 1.0 BACKGROUND NETWORK PROTOCOL FRAME STRUCTURE

DLL	Background Protocol		DLL
Header	**BPID**	**Protocol Data**	**Footer**
	1 byte	2 byte	2 byte

network protocols can be defined, currently only the D7A Advertising Protocol (D7AAdvP) is described in the specification. It is used for rapid ad-hoc group synchronization. The generic frame structure of the Background Network Protocol is shown in Table III. For the D7AAdvP, the protocol data consists of a 2 byte estimated time of arrival (ETA) in ticks. Ticks is the timing used in D7AP, 1 tick equals 2^{-10} seconds. The D7AAdvP is used by a device which wants to wake up one or more nodes as described in Subsection III-C.

The foreground network protocol structure is shown in Table IV. In D7AP 1.0 hopping is supported. Because of the larger range (resulting from the lower frequency used) many applications however can be solved using a less complex star topology. By default 2-hop communication is supported but data fields to enable multi-hop are present. Security is also organized by the network layer and is similar to the security of IEEE 802.15.4 [6] using AES-CBC for authentication and AES-CCM for Authentication and encryption.

D. Transport Layer

The Transport Layer defines the concept of request-response. It defines a method for acknowledging single and group requests. It provides the toolkit for minimizing the usage of D7AAdvP through a requester-controlled ad-hoc extension of the foreground scan. The D7A Transport Layer Protocol (D7ATP) organizes communication between Requester and one or many Responders in Transactions and Dialogs. In any single transaction one and only one requester sends a request during a Request Period which lasts for the time of transmission of one transport layer segment. The Response Period consists of any number of responses. Each responder is allowed to send at most one response. Transaction completes when all responders send their segments. A device in D7AP network can be involved in a Master Transaction which it initiates as a requester or a Slave Transaction in which it responds to a query. If subsequent queries need to be sent to the same group of devices this can be done in so called Dialog – a series of transactions separated by Access Period, when the requester executes the CSMA-CA routine.

The D7ATP segment structure is provided in Table V. The dialog ID and transaction ID are provided by higher layers of the stack. Dialog ID is a 1-byte unique value, while the transaction ID is a 1-byte incremental value that wraps at 255. The control byte of each D7ATP segment contains START and STOP bits. The requester initiates a dialog by setting the START bit to 1. Any subsequent response or a request in any subsequent transaction contain START bit set to 0. Upon ending the dialog the requester sets STOP bit to 1. The sole responder of a unicast request may initiate a new dialog directly after the termination of the old one by setting START control bit to 1 in the same transaction in which the requester sets STOP bit to 1. Devices not participating in a dialog discard any segments that are part of said dialog (START control bit set to 0), also responses not matching the transaction or dialog ID are discarded. This effectively prevents interruptions to ongoing dialogs.

The requester and the responders run a Dialog Timeout Timer (Dialog T_{TO}). The value of this timer can be controlled by the requester. The dialog expires when the dialog timeout timer elapses outside of a transaction. This timer can be reloaded at the end of any transaction if a new value is given in a Timeout Template or Transmission Timeout parameter of the Responder's Active Access Profile (T_C) if the current value of the timer is lower than T_C.

An Access Class (AC) is provided in NWL header during the Dialog. Only requesters can modify the AC during a dialog. The AC is an index defining Access Profile: scan type, CSMA-CA mode, number of sub-bands, subnet, scan automation period, transmission timeout period (T_C) and sub-bands. This means that the requester specifies the parameters of the dialog. During a dialog the channel is guarded by the Requester. Subsequent transmissions by requester or responses from a single responder need not use CSMA-CA procedure. In such case transmission is permitted unconditionally on any channel provided in the Requester Access Channel List.

E. Session Layer

All data exchanges with other DASH7 devices are executed within a Session. When a device is transmitting requests, which are received from the upper layer, it is executing a Master Session. Conversely, when a device is responding to requests received from the lower layer, it is participating in a Slave Session. Depending on the session type there are multiple states in which a session can be. For slave sessions the state is either active – meaning currently being executed – or done. A master session can be active or done as well, but can additionally be in an idle, dormant or pending state as well. A session in the pending state contains a group of requests which need to be transmitted as soon as possible. A dormant session on the other hand contains requests which need to be executed within a specified timeout period. After the timeout period has elapsed a dormant session is changed to a pending session. However, if the addressee of a dormant sessions starts a dialog before the timeout period is elapsed the requests in the dormant session can be transmitted by extending the dialog. This mechanism, which can be more efficient in certain situations, compared to actively waking up the device or constant polling of the endnode is explained above in Section III-E. Finally, a session in the idle state is currently not active. Session activation is prioritized in a way that sessions participating in an existing D7ATP dialog are executed first. This means that pending master sessions have to wait until active slave sessions are terminated. Dormant sessions that are activated before timeout, and thus participate

TABLE IV
D7AP 1.0 FOREGROUND NETWORK PROTOCOL FRAME STRUCTURE

DLL	D7A Network Protocol								DLL
Header	**Control**	**Hopping Control**	**Intermediary Access ID**	**Destination Access ID**	**Origin Access ID**	**Security Header**	**Payload**	**Auth. Data**	**Footer**
Length (bytes)	1	1	2/8	2/8	2/8	1/6	0-250	0-16	
		Optional						Opt.	
	Protectable				Protectable		Encryptable		

TABLE V
D7AP 1.0 TRANSPORT LAYER PROTOCOL SEGMENT STRUCTURE

DLL	NW	Transport Layer						NW	DLL
Header	**Header**	**Control**	**Dialog ID**	**Transaction ID**	**Timeout Template**	**Acknowledge Template**	**Payload**	**Footer**	**Footer**
		1 byte	1 byte	1 byte	0-1 byte	0-239 byte	0-239 byte		

in an extended dialog, also have priority over pending master sessions.

The D7AP Session Protocol (D7ASP) receives ALP Commands (see IV-G) from the upper layer application, and is responsible for transmitting the requests and meeting the specified QoS requirements. As part of the QoS strategy one can choose between different acknowledgment response modes: ”none”, ”all-cast” and ”any-cast”. In the ”any-cast” mode the request is acknowledged if a response is received to our request. This is useful for Sensor-to-Cloud cases where we just want to ensure our request is received by at least one gateway. ”All-cast” mode, on the other hand, returns all responses received during the dialog period to the application layer. Finally, when the response mode is ”none” the request is acknowledged if it was successfully transmitted after performing the CSMA-CA process, thus providing no reception guarantees. The QoS strategy also defines the maximum amounts of retransmissions to occur in case a required acknowledgment is not received. Per unique combination of Addressee and QoS parameters a FIFO is created in which a set of requests can be pushed, before being transmitted as separate requests being part of one dialog. The flushing of the FIFO is activated by the upper layer application or by the session layer itself for dormant sessions which either have expired or are activated because of the dialog extension of an active slave session with the requested addressee. Upon flushing a FIFO all queued requests will be transmitted by D7ATP using the same dialog ID and incrementing request IDs. For efficiency reasons, acknowledgments are not transmitted per received request but instead per group of requests within one dialog. Moreover, acknowledgments are explicitly requested by the requestor, who thus controls when to pause transmitting requests from the FIFO and instead collect acknowledgment information from the responders first. In a reply to this the responders each send their ack templates which is a bitmap showing the acknowledged request IDs within the dialog. Using this information the requestor can mark requests in the FIFO as completed, decide to retry transmission or drop requests when exceeding the retry counter. The flushing process is finished if all requests are either successfully completed or dropped, after which this is reported to the upper layer together with all received responses. Finally, the session layer might drive a power auto-scaling feature where the transmit power is dynamically adjusted based on the perceived link quality. This is entirely optional however and the specification does not define the algorithm to use, but considers this to be implementation specific.

F. Data Elements

D7AP specifies structured Data Elements, or files, which are managed in a filesystem together with their associated properties. The filesystem contains both D7A system files as well as user files. Files in the filesystem can be read from, written to or executed. Files are identified by a 1 byte file ID, where files 0x00 to 0x3F are reserved for system files. The system files, of which the content is strictly defined by the specification, contain configuration settings used by the D7A stack and describe the capabilities and status of the device. Since all files, including system files,are adaptable over the air interface (given the correct authentication) storing configuration parameters imply the behavior of the network can be changed after roll-out. The user file IDs and content are free to be chosen by the implementor. As already stated each file has associated properties like permissions, storage class and D7AActP (D7A Action Protocol, see IV-G) settings. The permission properties specify who can read, write and run the file and whether the file’s content is encrypted. D7AP defines 3 users: an unauthenticated guest user, a regular authenticated user and a root user which is either the device itself or an external user authenticated with the root key. The storage class describes the persistent option per file which can vary from permanent – meaning the file’s content is always available in memory for read and write operations – to transient – meaning the file’s content is not actually stored in memory and cannot be read back. Other storage class types are volatile and restorable, where the contents of a volatile file is stored in volatile memory and lost on power off. A restorable file also is read from volatile memory but snapshots can be written to persistent memory, an retrieved from memory in case of power

off. Finally, the last set of file properties refer to configuration of D7AActP. When D7AActP is enabled for a certain file, the system will trigger the execution of the ALP actions (see IV-G) specified in the configured action file. The triggering can be configured to only happen as a result of specific file actions, like read or write.

G. Application Layer Programming Interface

The Application Layer Programming Interface (ALP) is a generic API to manage the Data Elements of a D7AP node or a network of nodes through the usage of ALP Commands. While ALP Commands can be used to interact with a node's filesystem over any interface, for example using a UART or NFC connection, the obvious interface is the D7ASP (as described here IV-E) which enables communication with other D7AP nodes. Moreover, issuing ALP Commands to manage Data Elements using D7ASP is the only way to use D7AP to build an application. An ALP Commands is composed of one or more ALP Actions. The specification defines actions for reading data files, write file properties, querying, creating new files, authentication, execute file, etc. Depending on the action an operand may be required, for example reading file data requires a file data request operand containing the file ID, offset and length to read. A special action type is the query action. Queries allows to conditionally execute an action or group of actions. The condition is an arithmetic or string comparison between data in a file and supplied data or data in another file. Furthermore, conditions in a query can be chained together using logic operators. This mechanism allows addressing nodes based on business logic rules instead of only by ID, and allows to view a network of D7AP nodes as a distributed database which can be queried. Instead of interrogating a network using an ALP Command we can pre-register the same command on the nodes, as explained already in III-D, thereby configuring a dynamic push behavior. To conclude this section we describe the permission behavior of the ALP. By default, the application running on the device itself has root permissions, while commands received via a physically attached interface are executed with user level permissions. Finally, commands received over the air interface default to guest level permissions. The permission level can raised however by embedding an authentication action in the command. The permission is then raised for all subsequent actions in the same command, and dropped to the default level again after completion of the command.

V. Conclusion

In this paper we gave a brief overview of version 1.0 of the DASH7 Alliance Protocol. We showed how D7AP provides a complete network stack supporting multiple communication paradigms, which enables to implement many use cases in an efficient way. The concept of the filesystem which can be remotely managed allows for a great flexibility both during design phase and after roll-out. It contains, next to application specific data, also the network configuration parameters. Using the less congested sub-1 GHz bands, which provide increased range and permeability compared to 2.4 GHz, allows implementing many use cases with a star or tree network layout, avoiding the more complex multi-hop systems.

References

[1] H. Lehpamer, *RFID Design Principles*. Artech House, 2012.

[2] 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*, vol. Volume 2013, no. Article ID 870430, p. 9, 2013. [Online]. Available: http://www.hindawi.com/journals/ijdsn/2013/870430/

[3] *DASH7 Alliance Wireless Sensor and Actuator Network Protocol v1.0*, DASH7 Alliance Std., 05 2015. [Online]. Available: http://www.dash7-alliance.org

[4] *ISO/IEC/IEEE 21451-7: Information technology – Smart transducer interface for sensors and actuators – Part 7: Transducer to radio frequency identification (RFID) systems communication protocols and Transducer Electronic Data Sheet (TEDS) formats*, ISO/IEC/IEEE Std., 2011.

[5] *ISO/IEC 15963:2009 - Radio frequency identification for item management – Unique identification for RF tags*, ISO/IEC Std. 15 963, 2009.

[6] N. Sastry and D. Wagner, "Security considerations for ieee 802.15.4 networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, ser. WiSe '04. New York, NY, USA: ACM, 2004, pp. 32–42. [Online]. Available: http://doi.acm.org/10.1145/1023646.1023654