Low-cost Device Implementation of a Topology Independent MAC (TiMAC) Policy

Asterios Papamichail, Vasileios Dragonas,
George Koufoudakis, Konstantinos Oikonomou
Dept. of Informatics, Ionian University
Corfu, Greece
Email: {p15papa1, vdrag, gkoufoud, okon}@ionio.gr

Georgios Tsoumanis
Dept. of Informatics & Telecom.
University of Ioannina
Arta, Greece
Email: gtsoum@teiep.gr

Ioannis Stavrakakis
Dept. of Informatics & Telecom.
Nat. and Kap. University of Athens
Athens, Greece
Email: ioannis@di.uoa.gr

***Abstract*—The emerging new paradigm under the upcoming 5G communication paradigm is expected to support the multitudes of low-cost devices and an all-increasing number of new applications. Considering time-constrained applications (e.g., safety applications in vehicular networks) TDMA-based medium access control (MAC) policies, like those independent of the underlying topology (e.g., TiMAC) need to be revisited and now implemented in already existing low-cost devices. For these purposes, here TiMAC is considered and it is implemented in low-cost devices using 433 MHz RF modules. Even though at this stage the implementation is limited by synchronization issues and a small number of nodes, the obtained experimental results demonstrate the potential for employing TDMA-based MAC policies for safety applications in low-cost devices.**

***Index Terms*—Implementation, IoT, low-cost, time constraints, medium access control, topology independent**

I. INTRODUCTION

In the emerging new network paradigm as envisioned by 5G [1], numerous different devices are expected to be employed (e.g., 25 billion are expected until 2020 [2]) in order to support everyday activities. The Internet of Things (IoT) is expected to be an integral part of everyday life supporting a wide diversity of applications. These devices are expected to be of low cost, of small energy consumption, capable of communicating wirelessly, of sufficient computational capabilities and in most of the cases to be able to satisfy certain time constraints. For example, for safety applications in vehicular networks [3] it is expected the various devices to be able to communicate under certain time bounds.

This need – for time-constrained wireless communication of low-cost devices – has initiated a revisit of Time Division Multiple Access (TDMA) medium access control (MAC) policies that are inherently capable of meeting time constraints when compared to the contention-based MAC policies like Carrier Sense Multiple Access (CSMA) [4]. There are many benefits when using a TDMA approach (e.g., no need for the nodes to listen to the medium prior to transmitting [5], deterministic guarantees of throughput, etc.). On the other hand, certain challenges are introduced when a TDMA scheme is applied. One of them is synchronization among the various nodes [6] (since clocks do deviate as time passes [7], [8]) and another one is the assignment of *time-slots* to the network nodes within a TDMA *frame*.

In this paper, the idea of a TDMA-based topology independent medium access control (TiMAC) is adopted – as described in [9] – that allows for at least one successful transmission per frame independently of the underlying topology assigning time-slots based on properties of Galois polynomials presented in [9] and explained later in section II. This particular property allows for the use of TiMAC in applications of tight delay constraints. To implement this protocol, low-cost devices (i.e., Arduino [10]) are employed in the work presented here along with RF modules as 433 MHz[11], [12] that have no MAC implemented in order to create a multi-hop wireless network (more details regarding the low-cost devices are given later in Section III).

The challenge met in this paper using the previously mentioned low-cost devices is twofold: (i) to implement TiMAC following the rules (and properties of Galois polynomials) as described in [9]; and (ii) to synchronize nodes in a multi-hop decentralized manner. As it is shown later in the paper, TiMAC is successfully implemented (one successful transmission per frame is shown that is guaranteed) and a decentralized approach is followed regarding synchronization that successfully synchronizes nodes in a multi-hop manner.

The experiments conducted in this paper and presented in Section IV are limited to five only nodes in a full graph topology configuration that allows for giving a detailed insight of the particulars of the implementation (e.g., the Galois polynomials assigned to nodes). The synchronization part is implemented in a multi-hop manner despite the single hop topology, is demonstrated to be successful and ready for implementation in a real multi-hop topology. Still, its current performance is limited by prolonged time guard periods and therefore part of the future work will involve improving the synchronization mechanism (reduce time guard periods, exploit acknowledgments, etc.) and experimentation using a large set of nodes in a multi-hop environment. Still, despite implementation-specific limitations, the implemented system in this paper is capable of demonstrating the potential of implementing topology independent MAC policies like TiMAC for time-constrained applications in low-cost devices.

A brief description of TiMAC is included in Section II and details about the low-cost device, the 433 MHz RF modules and the TDMA synchronization approach are given in the system's description Section III. The experimental setup (i.e., the topology and the various parameters) along with the experimental results are presented in Section IV and the conclusions along with directions for future work in Section V.

II. TiMAC DESCRIPTION

The TiMAC policy operates under certain assumptions regarding the total number of nodes in the network (denoted by N) as well as the maximum number of neighbors (denoted by D) that a node may have. TiMAC relies on TDMA having a fixed-length frame for all nodes and the time-slot duration is considered long enough for any data packet to be transmitted. Each node is assigned a subset of the frame's time-slots for transmitting according to arbitrarily assigned Galois polynomials as described next. This subset is unique for each node, ensuring that each of them transmits according to a unique repeating pattern. Furthermore, this uniqueness ensures that there will be at least one successful transmission for each node in every frame, even if the network load is increased (e.g., each node has always data packets available for transmission) [9], as explained later.

According to the Chlámtac-Farago algorithm [9] the number of network nodes N and the maximum number of neighbor nodes D can be used to derive the values of two new integer parameters, i.e., q (related to the size of the frame) and k (related to the degree of Galois polynomials of order q). More specifically, the frame size is q^2 (i.e., it consists of q^2 time-slots) and it is split into q sub-frames of q time-slots each. One slot of each of the q sub-frames is assigned to each node. Therefore, each node is allowed to transmit at q time-slots within a frame (one during each subframe).

The next step is to derive the particular time-slots that a certain node u is allowed to transmit. For this reason, each node u is assigned to a unique polynomial of degree k with coefficients from a finite Galois field of order q ($GF(q)$) and is represented as $f_u(x) = \sum_{i=0}^{k} a_i x^i$, where $a_i \in \{0, 1, 2, \ldots, q-1\}$. The slot assigned to each node u inside a subframe s is given from $f_u(s) \bmod q$ where $s \in \{0, 1, \ldots, q-1\}$ [9]. Apparently, the constraint for unique polynomials assigned for all nodes in the network, requires that $q^{k+1} \geq N$ (q^{k+1} corresponds to all possible polynomials of degree k when their coefficient belong to the $GF(q)$).

An integral property of this approach is that there is at least one successful transmission per frame (i.e., no collision takes place) even for the case of increased load conditions (e.g., all nodes always have data packets ready to be transmitted). Since each node transmits according to its assigned polynomial of degree k, there will be at most k simultaneous transmissions (thus collisions will occur) with any neighbor node since there are at most k common roots between two polynomials of degree k. Since the number of neighbor nodes is at most D, then there will be at most kD simultaneous transmissions. Given that each node is allowed to transmit at q time-slots, to guarantee at least one successful transmission, $q \geq kD + 1$. Both constraints, $q^{k+1} \geq N$ and $q \geq kD + 1$ are exploited by the algorithm in [9] to calculate q and k given N and D.

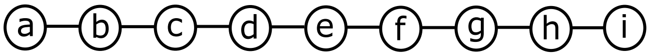

Fig. 1. A chain topology of nine nodes.

	Standard TDMA Frames										TiMAC TDMA Frame								
											Subframe 0			Subframe 1			Subframe 2		
Nodes	0	1	2	0	1	2	0	1	2	Polynomials	0	1	2	0	1	2	0	1	2
a	x			x			x			0x+0	x			x			x		
b		x			x			x		0x+1		x			x			x	
c			x			x			x	0x+2			x			x			x
d	x			x			x			1x+0	x				x				x
e		x			x			x		1x+1		x				x	x		
f			x			x			x	1x+2			x	x				x	
g	x			x			x			2x+0	x					x		x	
h		x			x			x		2x+1		x		x					x
i			x			x			x	2x+2			x		x		x		

Fig. 2. Standard and TiMAC TDMA Frame examples.

Assuming the chain topology depicted in Fig. 1, both the corresponding "standard" and the TiMAC TDMA frames are presented in Fig. 2. On the left of Fig. 2, the standard TDMA frame consisting of three slots is depicted (repeated three times) where each node has been assigned a fixed slot in a way that ensures collision-free communication between neighbors. On the right, a TiMAC frame is depicted consisting of three subframes where the slot assignment has taken place according to the corresponding arbitrarily assigned polynomials. Under TiMAC, topology changes may take place (assuming N and D remain as such) and still every node is guaranteed at least one successful transmission per frame whereas, under standard TDMA, special care has to be taken after a topology change as a slot reassignment may be necessary.

The next step is to implement this particular MAC policy in the low-cost devices that their characteristics are described in the following section.

III. SYSTEM DESCRIPTION

The low-cost devices that implement each node of the wireless network considered here have certain specifications regarding their operation that need to be taken into account in order to implement the previously described TiMAC policy. The particular information along with various implementation issues (e.g., TDMA synchronization) are presented next.

A. Low-cost device elements

Three essential elements have been used for each node device: the core part – that is an Arduino prototyping platform [10] – and the two RF modules operating at 433 MHz that are used for transmission and reception [11], [12].

Arduino [10] is an open source and low-cost electronics platform, easy to use and capable of hosting various components (for the case considered the RF modules). The particular board used for this paper is the Arduino Uno Rev. 3. It is a board built on top of the ATMEL ATMEGA328 microcontroller with a clock speed of 16MHz and 2KB of SRAM [13].

The wireless communication part was handled by short range RF modules operating in the license-free *Industrial, Scientific and Medical* (ISM) Band of 433.050 - 434.790 MHz. Two different components are used to realize transmission (Tx) and reception (Rx), i.e., RWS-371-V1.0.3 and TWS-BS-V1.0.3 [11], [12], respectively. As already mentioned these modules co-operate with Arduino and the wiring used in this paper is schematically illustrated in Fig. 3.

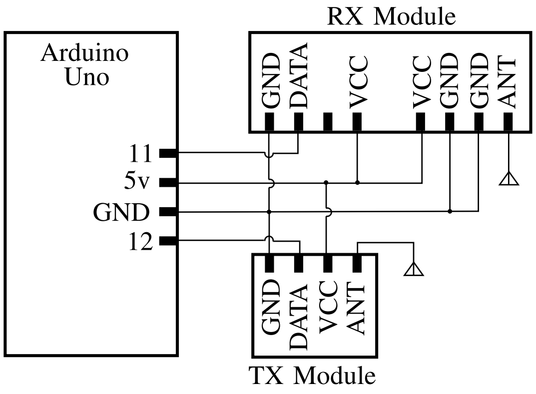

Fig. 3. The wiring between the Arduino and the RF modules that implements the device for both transmission (i.e, Tx-module data connected to pin 12) and reception (i.e, Rx-module data connected to pin 12). The antenna (ANT) along with the power supply pins (i.e., GND and VCC) are also depicted.

B. Message types

For the implementation of the TDMA protocol, the messages that were used for the purpose of this paper are `SYNC` and `DATA`. `SYNC` messages are sent by the nodes at the beginning of each frame, for time synchronization purposes. The payload of `SYNC` messages contains the time remaining until the first time slot. Any node that has data for transmission constructs a `DATA` message which is transmitted during the particular node's time slot. The existence of these two distinct message types is necessary for the smooth operation of the TDMA protocol. `SYNC` messages are processed at the beginning of each frame to synchronize the receiver's TDMA clock timer whereas, `DATA` messages are transmitted by the nodes during the particular time-slots that have been assigned to them.

C. TDMA parameters

The implementation of TDMA protocol on the ISM Band of 433.050 - 434.790 MHz, using low-cost devices introduces several limitations regarding frame timing parameters. In particular, nodes' internal clock may deviate significantly [14], the interference on the selected band may be high [15], [16], and the low-cost transmitter's accuracy and receiver's sensitivity may be reduced.

Considering the above limitations, each time slot (ts_i) of the TDMA frame lasts for 500ms, whereas a 300ms guard time (t_g) is introduced between them in order to avoid collisions due to clock drifts. Moreover, at the beginning of each frame, nodes are allowed to get synchronized for a time period of 2000ms (st). At the end of the frame, a frame guard period of 500ms (t_{fg}) is included, allowing for further node's operations (e.g., sensing, message processing) as depicted in Fig. 4.

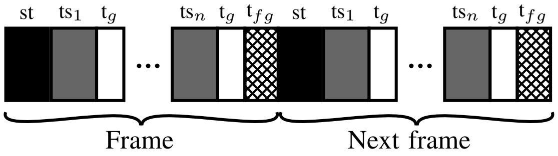

Fig. 4. The TDMA frame of the implemented MAC protocol. Each time slot (ts_i) ends with the guard period (t_g). The synchronization time (st) along with the frame guard period (t_{fg}) are also depicted.

D. Synchronization

In TDMA environments, nodes have to be aware of the time that the frame begins. Moreover, implementing TDMA in multi-hop network environments requires a (multi-hop) decentralized clock synchronization mechanism.

Several synchronization algorithms that employ MAC layer timestamping (e.g., [17], [18], [19]) have been proposed in the literature. Since this work focuses on the implementation of TiMAC in low-cost devices, the following simple synchronization mechanism is considered. Nodes are allowed to synchronize their clocks for a period of 2000ms at the beginning of each frame, by transmitting the previously described *SYNC* messages. More specifically, a reference node transmits a *SYNC* message to its neighbor nodes that includes the time remaining until the first time slot of the frame. Subsequently, every node that receives this message for the first time adjusts its clock accordingly and transmits a new *SYNC* message to its neighbor nodes.

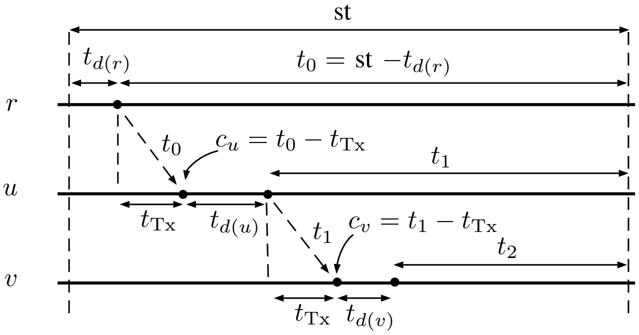

Fig. 5. Illustration of the synchronization mechanism within a synchronization period (st), among nodes r, u and v connected in a chain topology. Nodes wait for a certain amount of time (i.e., t_{dr}, t_{du} and t_{dv} respectively) before transmitting the time remaining until the first time slot (i.e., t_0, t_1 and t_2 respectively). The internal clocks c_u, c_v along with the transmission time t_{tx} are also depicted.

Fig. 5 depicts the synchronization mechanism between the reference node (r), and nodes u and v, connected in chain topology (i.e., r-u-v). At the beginning of the synchronization period (st), the reference node r waits for a certain amount of time t_{dr} and then transmits its clock $t_0 = st - t_{dr}$ to node u. When the message is received, node u subtracts the transmission time from t_0 and changes its clock accordingly (i.e., $c_u = t_0 - t_{tx}$). Subsequently, node u waits for a certain amount of time (i.e., t_{du}) and then transmits its clock (i.e., t_1) to node v. When the message is received, node v changes its clock (i.e., $c_u = t_1 - t_{tx}$), waits for a certain amount of time (i.e., t_{dv}) and transmits its clock to the next-in-line node (if any). Apparently, node u may also receive node's v *SYNC* message, but since one has already been received, this one may be ignored.

E. TiMAC Implementation

The implementation of TiMAC protocol is based on TDMA as described earlier. Five nodes are considered here, and it is assumed that they are part of a larger network of maximum $N = 25$ nodes where the maximum number of neighbors is $D = 4$. Based on those values, the corresponding values for k (i.e., $k = 1$) and q (i.e., $q = 5$) are derived [9]. Eventually, first-degree polynomials are going to be used for the slot allocation ($k = 1$), and the length of the TDMA frame will be $q^2 = 25$, divided into five subframes of $q = 5$ slots each. Each node is randomly assigned a first-degree polynomial with coefficients from $GF(5)$ and according to that polynomial, the particular time slot that each node is allowed to transmit within each sub-frame is derived. Thus, every node is allowed to transmit in five slots per frame.

IV. EXPERIMENTAL RESULTS

The experimental setup consists of five nodes in a full graph topology such that, each node has exactly four neighbors. Collisions are expected to occur. However, under the TiMAC policy, it is expected that at least one transmission per node per frame to be successful, i.e., free of collisions.

Note that transmissions may be unsuccessful for various other reasons. For example, there might be interference in this band, since the 433Mhz band is an ISM one. Special care was taken to conduct the experiments in isolated areas (e.g., the basement of the premises of the campus).

A. Methodology

Eleven different sets of polynomials are considered here that correspond to eleven different scenarios for the particular system. Consequently, different time-slots were assigned to each node per scenario and the obtained results, is the summation of transmissions of 100 frames. Heavy traffic conditions are assumed for the experimental setup, and therefore each node has always data ready for transmission at every time slot. For each node, a neighbor node was arbitrarily selected as the destination of its transmitted packets and remained fixed throughout the experiments.

The first set of polynomials was chosen in order to create a collision-free slot allocation such that, each node transmits packets in different time-slots than that of its neighbors. The rest ten polynomial sets were created randomly.

The obtained *throughput* is derived as the fraction of the average successful transmissions (i.e., total successful transmissions over the number of nodes) over the number of total time-slots i.e.,

$$\text{throughput} = \frac{\text{total successful transmissions}}{\text{total nodes} \times \text{total time slots}}.$$

For the period of 100 frames, each node may transmit during up to 500 time-slots whereas, the number of total time-slots is 2500. Consequently, the maximum network throughput is given by,

$$\text{capacity} = \frac{500}{2500} = 0.2.$$

Throughput as large as capacity is not expected to be achieved due to various reasons like the channel interference, the transmitter's accuracy and the receiver's sensitivity, etc., as already mentioned.

On the other hand, TiMAC guarantees that one successful transmission per node per frame will be successful. Consequently, the minimum network throughput is given by,

$$\text{minimum throughput} = \frac{100}{2500} = 0.04.$$

B. Results

Figure 6 depicts the derived throughput for the eleven polynomial sets. It is observed that in all cases it is higher than the minimum value guaranteed by the TiMAC policy (i.e., 0.04). It is also observed that the first polynomial set has the highest value for throughput; however, it is smaller than the maximum value (i.e., 0.2). This is expected since, on the one hand, the particular set of polynomials results in a collision-free slot allocation and on the other hand, the channel interference, transmitter's accuracy, and receiver's sensitivity may result in unsuccessful transmissions.

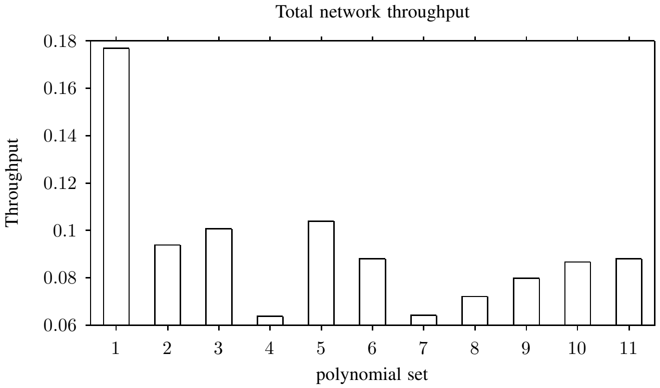

Fig. 6. Throughput corresponding to various polynomial sets. In all cases, the throughput lays between the expected maximum (i.e., 0.2) and minimum (i.e., 0.04) values.

Figure 7 depicts the average successful transmissions per frame per node for the eleven polynomial sets. The particular results confirm the TiMAC policy's lower bound of one successful transmission per frame per node. For the first polynomial set, significantly more successful transmissions than the other ones are observed. However, the are less than the maximum theoretical value of five, which is the successful transmissions per node per frame if no packets are lost for the duration of the 100 frames.

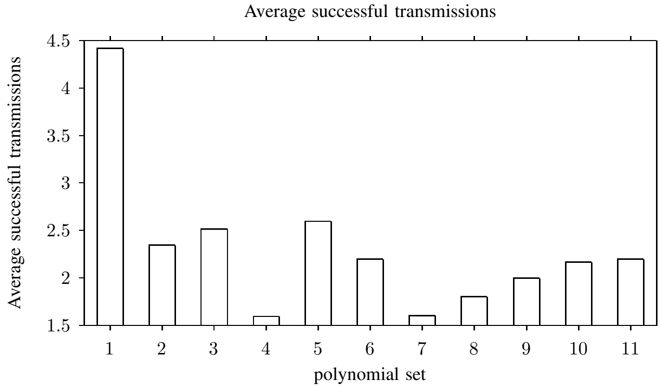

Fig. 7. Average number of successful transmissions per frame per node corresponding to various polynomial sets. In all cases, the number of successful transmissions is between the expected maximum (i.e., 5) and minimum (i.e., 1) values.

V. CONCLUSIONS AND FUTURE WORK

An implementation of the TiMAC policy is presented in this paper based on the arduino microcontroller and low-cost RF modules operating in the 433MHz ISM band. A multi-hop decentralized clock synchronization mechanism is adopted for the implementation of the underlying TDMA protocol. The particular low-cost devices pose several limitations regarding clock accuracy, transmission accuracy, and reception sensitivity.

The experimental system consisted of five nodes in a full graph topology such that, each node had exactly four neighbors and the results were collected for eleven scenarios using different sets of polynomials. The experimental results confirm the analytical findings of TiMAC. It is shown that each node has at least one successful transmission within a frame and the total network throughput lays within the minimum and maximum analytical expressions.

Part of the future work in the area will involve improving the synchronization mechanism and conducting experiments using more than five nodes in a multi-hop environment. Still, despite these implementation-specific limitations, the particular system demonstrates the potential of implementing time independent MAC policies like TiMAC for time-constrained applications in low-cost devices.

ACKNOWLEDGMENT

This work was supported in part by project "A Pilot Wireless Sensor Networks System for Synchronized Monitoring of Climate and Soil Parameters in Olive Groves," (MIS 5007309) which is partially funded by European andNational Greek Funds (ESPA) under the Regional Operational Programme "Ionian Islands 2014-2020." In addition, this work was supported in part by the European Commission as part of the ReCRED project (Horizon H2020 Framework Programme of the European Union under GA number 653417), by the Chair of Excellence UC3M - Santander Program and by the National and Kapodistrian University of Athens (S.A.R.G.).

REFERENCES

[1] A. Osseiran, J. F. Monserrat, and P. Marsch, *5G mobile and wireless communications technology*. Cambridge University Press, 2016.

[2] G. Intelligence, "Definitive data and analysis for the mobile industry," *GSMAintelligence. com*, 2016.

[3] V. Dragonas, K. Oikonomou, K. Giannakis, and I. Stavrakakis, "A disjoint frame topology-independent tdma mac policy for safety applications in vehicular networks," *Ad Hoc Networks*, vol. 79, 06 2018.

[4] M. Hadded, P. Mühlethaler, A. Laouiti, R. Zagrouba, and L. A. Saïdane, "Tdma-based MAC protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2461–2492, 2015.

[5] S. C. Ergen and P. Varaiya, "Tdma scheduling algorithms for wireless sensor networks," *Wirel. Netw.*, vol. 16, no. 4, pp. 985–997, May 2010. [Online]. Available: http://dx.doi.org/10.1007/s11276-009-0183-0

[6] F. Wang, P. Zeng, and H. Yu, "Slot time synchronization for tdma-based ad hoc networks," in *2008 International Symposium on Computer Science and Computational Technology*, vol. 2, Dec 2008, pp. 544–548.

[7] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Networks (Elsevier*, vol. 3, pp. 281–323, 2005.

[8] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[9] I. Chlamtac and A. Faragó, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 23–29, 1994.

[10] M. Banzi and M. Shiloh, *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2014.

[11] *Wireless Hi Power Transmitter Module (RF ASK)*, WENSHING ELECTRONICS CO.,LTD, 10 2010, v1.0.3.

[12] *Wireless Hi Sensitivity Receiver Module (RF ASK)*, WENSHING ELECTRONICS CO.,LTD, 12 2008, v1.0.3.

[13] *Arduino*, Arduino LLC, 9 2010, rev3.

[14] C. Guo, J. Shen, Y. Sun, and N. Ying, "Rb particle filter time synchronization algorithm based on the dpm model," *Sensors*, vol. 15, no. 9, pp. 22 249–22 265, 2015.

[15] "Ieee recommended practice for powering and grounding electronic equipment," *IEEE Std 1100-2005 (Revision of IEEE Std 1100-1999)*, pp. 1–703, May 2006.

[16] D. Robinson, T. A. Wysocki, V. W. Smith, and K. Popovski, "Background radio frequency interference measurements for wireless devices in the electricity supply industry," 2005.

[17] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 39–49.

[18] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 138–149.

[19] K. Skiadopoulos, A. Tsipis, K. Giannakis, G. Koufoudakis, E. Christopoulou, K. Oikonomou, G. Kormentzas, and I. Stavrakakis, "Synchronization of data measurements in wireless sensor networks for iot applications," *Ad Hoc Networks*, 2019.