

Article

Implementation of a Topology Independent MAC (TiMAC) Policy on a Low-Cost IoT System

Georgios Tsoumanis ^{1,*}, Asterios Papamichail ^{2,†}, Vasileios Dragonas ^{2,†},
George Koufoudakis ^{2,†}, Constantinos T. Angelis ^{1,†} and Konstantinos Oikonomou ^{2,†}

¹ Department of Informatics & Telecom., University of Ioannina, 45110 Arta, Greece; kangelis@uoi.gr

² Department of Informatics, Ionian University, 49100 Corfu, Greece; aspapa@ionio.gr (A.P.);
vdrag@ionio.gr (V.D.); gkoufoud@ionio.gr (G.K.); okon@ionio.gr (K.O.)

* Correspondence: gtsoum@uoi.gr

† This paper is an extended version of our paper published in the proceedings of the Third International Balkan Conference on Communications and Networking (BalkanCom 2019), Skopje, North Macedonia, 10–12 June 2019.

‡ These authors contributed equally to this work.

Received: 31 March 2020; Accepted: 3 May 2020; Published: 11 May 2020



Abstract: The emerging new paradigm under the fifth generation of wireless communications technologies (5G) and high expectations for massively expanding today's Internet of Things (IoT) under 5G, are expected to support a large plurality of low-cost devices for an all-increasing number of new IoT applications. Many emerging IoT applications are going to take advantage of techniques and technologies that have high demands from low-cost devices in terms of processing large amounts of data and communication. For example, in systems based on fog computing technology, low-cost devices have to assign some of their limited resources for processing purposes. Considering the drawbacks emerging from using low-cost devices and the fact that many applications are in need for time-constrained approaches, TDMA-based *Medium Access Control* (MAC) policies need to be revisited and implemented in low-cost devices of today. In this sense, a policy independent of the underlying topology, TiMAC policy, is considered here and is implemented in low-cost devices using 433 MHz RF modules. Even though 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 on IoT systems consisting of low-cost devices.

Keywords: implementation; IoT; low-cost; time constraints; medium access control; topology independent

1. Introduction

Recent technological advancements in IoT, as well as the emerging new network paradigm as envisioned by 5G [1,2], have been the motivation for employing numerous different devices in order to support everyday activities. IoT, firstly introduced in 1999 [3], has been widely adapted to a variety of fields, such as environmental monitoring, infrastructure management, and home automation [4], as a concept of making the devices “smarter”—thus less dependent on human intervention.

By the end of 2020, 25 billion IoT devices are expected to have been employed [5] worldwide. The growth of IoT and the increase in the supported devices lead to the generation of large amounts of data that have to be processed, stored, and transmitted by the different devices. In order for this data to be processed, stored, and transmitted, massive computing resources, storage space, and communication bandwidth are needed. In this sense, many works have focused their attention on tackling the above constraints. For example, computing tasks in IoT devices regarding the generated data can be assigned to cloud servers, since the computing power on the cloud outclasses the capability of the IoT devices [6].

In addition, in the post-cloud era arriving, many applications will also be deployed at the *edge* of a system [7], so as to bring services closer to the IoT devices by pooling the available computing, storage, and networking resources at the edge of the network [4].

Regarding the devices composing an IoT system, they have to be of low cost and consume as little energy as possible, in order to support implementation on large-scale and multi-hop environments. In addition, sufficient computational capabilities are required for supporting the emerging applications and techniques, such as acting as edge devices in an edge-computing based system, where many processes occur directly on the IoT devices, rather than on the cloud.

MAC policies, in IoT and networks overall, have a significant impact on the devices' characteristics described above. For example, MAC policies control the radio transmitter which is the most energy demanding part of an IoT device [8]; thus, the overall energy consumption and lifetime of the devices are largely determined by the MAC policy implemented. Apart from energy conservation (which is the case in many works in the literature [9,10]), the role of MAC policies is to decide how the devices will access the shared medium, ensuring that only one node may access the channel at a time.

Regarding the aforementioned devices' access to the shared medium, in many systems, specific time constraints have to be satisfied. For example, for safety applications in vehicular networks [11], it is expected that various and different devices can communicate and transmit their data under certain time bounds. Combining the above with the requirement for low-cost devices, as mentioned earlier, there is a need for more research on time-constrained wireless communication of low-cost devices. This need has initiated a revisit of Time Division Multiple Access (TDMA) MAC policies that are inherently capable of meeting time constraints when compared to the contention-based MAC policies, such as Carrier Sense Multiple Access (CSMA) [12] policies. Furthermore, in a system under a TDMA policy, there is no need for the nodes to listen to the medium prior to transmitting [13], while at the same time TDMA's deterministic approach guarantees a minimum performance level in terms of throughput. On the other hand, specific challenges arise when TDMA policies are applied, such as the need for continuous synchronization among the devices [14]. The latter drawback's difficulty to tackle lies both on the aforementioned extensive data generated by IoT devices and on the deviation occurring on the different devices' clocks as time passes [15,16]. It is argued that this problem renders the use of TDMA policies impractical. However, there are network scenarios where the synchronization problem is not present (e.g., the use of the already present GPS receivers in Vehicular Networks) and new ways may be discovered as technology advances. In addition to the synchronization problem, *time-slots* assignment to the network nodes within a TDMA *frame* is also a prerequisite for a TDMA policy to work properly.

In this paper, the idea known from the literature TDMA-based policy is revisited. Then, this policy's theoretical expectations are evaluated for their effectiveness on a system consisting of low-cost devices while a decentralized approach is followed regarding synchronization. More specifically, the proposed by Chlamtac and Faragó in [17] TDMA-based topology independent medium access control (to be referred to as TiMAC in the sequel) policy is adopted. TiMAC policy's primary goal is to allow for at least one successful transmission per node per frame, independently of the underlying topology. In order to achieve this goal, TiMAC's time-slot assignment method is based on the properties of Galois polynomials, also presented in [17]—thus allowing for its implementation on applications of tight delay constraints. The system considered here for implementing TiMAC on consists of devices of low-cost boards (i.e., Arduino [18]) carrying simple RF modules at 433 MHz [19,20]. Note that, in their initial condition, devices are free of any software or coding installed on them (e.g., MAC).

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 [17]; and (ii) to synchronize the devices (also referred as nodes in the sequel) in a multi-hop decentralized manner. As it is shown, TiMAC is successfully implemented (i.e., one successful transmission per frame is shown that is guaranteed) on the low-cost system and the decentralized synchronization approach manages to successfully synchronize the nodes in a multi-hop manner.

The above contributions rely on experiments conducted and presented in Section 5. Note that the experiments mentioned above are limited to only five nodes in a full graph topology configuration, thus allowing for getting a detailed insight of the particulars of the implementation (e.g., the Galois polynomials assigned to nodes). As for the synchronization part of the experiments, it is implemented in a multi-hop manner despite the single-hop topology and time periods dedicated to synchronization are employed between time frames. As it is demonstrated, the nodes are successfully synchronized, showing that this method can be easily implemented in a more realistic and multi-hop topology.

The rest of this paper is organized by including past related work in Section 2 and a brief description of TiMAC in Section 3. Details about the low-cost device, the 433 MHz RF modules and the TDMA synchronization approach are given in the system's description Section 4. The experimental setup (i.e., the topology and various parameters) along with the experimental results are presented in Section 5 and the conclusions are drawn in Section 6.

2. Past Related Work

MAC policies for IoT have been the case in many works over the past few years. Oliveira et al., in their 2019 work [21], survey MAC policies used in IoT, providing a detailed description of each one. More specifically, they categorize the reviewed policies in short and long-distance coverage. Under the latter categorization, they survey most of the known policies used by, e.g., Radio Frequency Identification (RFID), Bluetooth Low Energy (BLE), Wireless Highway Addressable Remote Transducer Protocol (Wireless-HART), Long Range Protocol (LoRa), and many more.

Like Oliveira et al., Kumar et al., in their work in [22], survey MAC policies in IoT. The authors there categorize the considered MAC policies according to the channel access method each one uses. In addition, they give the essential properties of MAC policies for IoT and the common causes of energy consumption. Finally, they address some pros and cons for each surveyed policy.

Other surveys, more dedicated to the applications each IoT system is designed for, can be found in the literature. For example, Sherazi et al., in [10], provide a detailed analysis of MAC policies exclusively for energy harvesting IoT systems. In contrast, a review and evaluation of MAC policies for satellite IoT systems using nano-satellites are given in [23] by Ferrer et al.

TDMA policies for IoT, hybrid policies that consider the cooperation of TDMA with other channel access methods, and improvements for TDMA policies have been the case in several works in the literature over the past five years. A distributed and adaptive hybrid medium access control (DAH-MAC) is proposed in [24] by Ye and Zhuang. In the policy considered there, a distributed TDMA approach is employed for voice packet transmissions to guarantee a voice packet loss rate bound, while truncated CSMA/CA (TCSMA/CA) is used for data nodes to access the channel. The same authors in [25] propose a token-based adaptive medium access control (TA-MAC) scheme for a two-hop Internet of Things (IoT)-enabled mobile ad hoc network. TA-MAC bases its operation on super-frames created by the devices and these super-frames' duration, along with the number of token rotation cycles, are optimized and adapted to the network's traffic load, so as to achieve a consistently minimal average end-to-end packet delay.

Another hybrid approach is given in [26] by Shahin et al. There, a hybrid slotted-CSMA/CA-TDMA (HSCT) policy is proposed, where a contention-based slotted-CSMA/CA period (SCP) divided into multiple CSMA/CA access windows is assumed, along with a contention-free slotted-TDMA period divided into multiple slots. Both CSMA/CA and slotted-TDMA are cooperating in order to achieve the registration of machine-to-machine (M2M) communications in large scale environments where many devices try to access a single and centralized access point.

An algorithm for fair resource allocation in TDMA-based distributed sensor networks for IoT is proposed in [27] by Kim et al. The authors of the latter paper implement a multi-hop desynchronization (DESYNC) algorithm for tackling the hidden-node problem in multi-hop TDMA-based networks, by proposing a new frame structure and a procedure dedicated to approaching the problem.

Scheduling in TDMA-based IoT systems is studied in [28] by Li et al. The authors in this paper propose a distributed TDMA slot scheduling algorithm that takes into account nodes' energy and the network's topology. In this approach, the residual energy of nodes and the topology knowledge about the number of nodes' neighbors play a significant role in the time slot allocation, in providing higher priority in nodes of lower residual energy and more neighbors than the other nodes. In the same spirit, the authors in [29] introduce a distributed TDMA scheduling algorithm based on exponential backoff rule, the nodes' residual energy, and the number of neighbors.

All works given above regarding TDMA policies in IoT are some of the newest in the literature (i.e., 2015–2020). Interestingly, in the works referenced above, there are two similarities in all of them: (i) All of the policies and algorithms proposed have been tested on simulators, rather than on real devices; and (ii) synchronization of nodes is not the case in any of them. At the same time, in most, it is either assumed that GPS infrastructure is used by the nodes (thus synchronization is achieved through them), or centralized approaches are assumed. As a result, there is no knowledge on how these approaches would work on low-cost systems, where fewer options are available (e.g., no GPS can be implemented due to its high cost).

The current work is based on deploying a known policy (TiMAC) on a low-cost IoT system. In addition, considering the drawbacks of the low-cost system (further explained later), a decentralized approach for the synchronization of nodes in such systems is given. TiMAC was firstly introduced by Chlamtac and Faragó in [17] and it has been further extended and implemented in various applications. For example, the authors [11] consider TiMAC in their paper for vehicular networks, while they also extend TiMAC to one that considers disjoint frames depending on the vehicle's direction of movement.

3. The TiMAC Policy

TiMAC is a TDMA-based policy, the latter policies dividing the time into slots and each node is able to access the channel at a given time slot. Note that, in TDMA, all nodes share the same frequency channel; thus, they have to be synchronized for ensuring that interference will not be experienced during transmissions. In this sense, under TiMAC, a fixed-length frame is assigned to all nodes, while the time-slot duration is considered to be long enough for a transmission of a data packet to take place. The next step is to assign to each node a subset of the previously assigned frame's time-slots, during which a node is allowed to transmit its data packet. The latter time-slot assignment takes place according to arbitrarily assigned Galois polynomials [30] as described next. The goal here is to ensure a unique subset for all nodes—thus further ensuring that each node will transmit according to a unique repeating pattern. As a result, at least one transmission will be successful for each node during every frame, even if the network load is increased (e.g., each node always has data packets available for transmission) [17].

As already mentioned, under the TiMAC policy, nodes are assigned time-slots according to arbitrarily assigned Galois polynomials, the latter procedure being derived by the Chlamtac–Faragó algorithm [17]. In order for the TiMAC policy to exploit the Chlamtac–Faragó algorithm and operate, certain assumptions have to be made regarding the total number of nodes composing the network, as well as the maximum number of neighbors each node can have. Let N denote the number of network nodes and D the maximum number of neighbors, respectively. N and D are used by the algorithm in order to derive the values of two new (integer) parameters, the first related to the size of the frame and denoted by q , and the second related to the degree of Galois polynomials of order q and denoted by k . More specifically, the frame size consists of q^2 time-slots (i.e., the frame size is q^2), each one split into q sub-frames of q time-slots. Each node is assigned one slot from each one of the q sub-frames and, therefore, each node is allowed to transmit at q time-slots within a frame.

Having made the above assumptions, the next step is to derive the particular time-slots during which it is allowed for a certain node u to transmit its data packet. For deriving the latter time-slot,

each node u is assigned a unique polynomial of degree k with coefficients from a finite Galois field of order q ($GF(q)$), to be represented as

$$f_u(x) = \sum_{i=0}^k a_i x^i$$

where $a_i \in \{0, 1, 2, \dots, q - 1\}$. In order to find the slot assigned to each node u inside a subframe s , $f_u(s) \bmod q$ is calculated, where $s \in \{0, 1, \dots, q - 1\}$ [17]. It is reasonable to suppose that, in order to achieve the constraint for unique polynomials assigned to all nodes in the network, it is required 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, in each frame, at least one transmission will be successful for each node even for the case of increased traffic load conditions. For example, even if all nodes always have data packets ready to be transmitted, each node will have at least one collision-free slot during the frame. Taking into account that 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 . Considering that the maximum number of neighbor nodes is D , then at most kD simultaneous transmissions are expected to take place. In addition, given that each node is allowed to transmit at q time-slots, then $q \geq kD + 1$ so as to guarantee at least one successful transmission. Both constraints, $q^{k+1} \geq N$ and $q \geq kD + 1$, are exploited by the Chlámtac–Farago algorithm to calculate q and k given N and D .

In order to gain more insight into TiMAC, in Figure 1a, an example chain topology is depicted. Regarding the depicted topology, both the corresponding “traditional” and the TiMAC TDMA frames are presented in Figure 1b. The traditional TDMA frame is given on the left side of Figure 1b. It consists of three slots that are repeated three times and each node has been assigned a fixed slot in a way that ensures a collision-free communication among neighbors. On the other hand, a TiMAC frame is depicted on the right side of the figure. It consists of three subframes on which the slot assignment has taken place in accordance with the corresponding arbitrarily assigned polynomials. It is observed here that, when the nodes operate under TiMAC, every node is guaranteed at least one successful transmission per frame, even in the cases where topology changes occur. However, when traditional TDMA is chosen, 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 a system that consists of low-cost devices and evaluate its performance. The proposed system’s characteristics are given in the following Section 4.

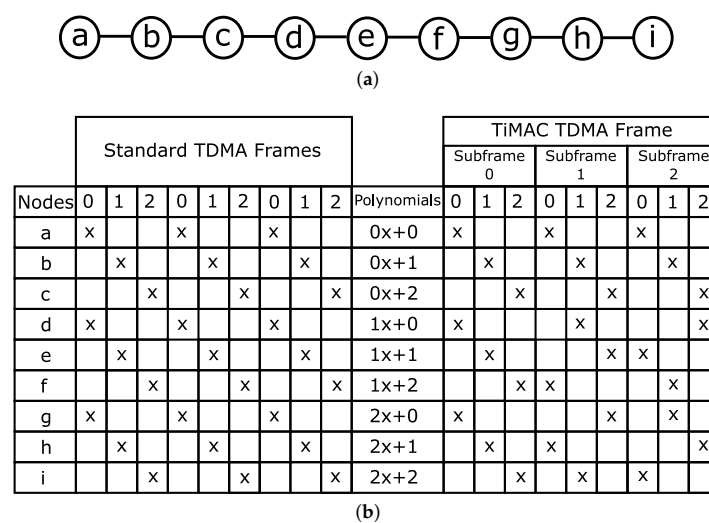


Figure 1. (a) a chain topology of nine nodes; (b) traditional and TiMAC TDMA Frame examples.

4. The Proposed System

A network consisting of low-cost devices is assumed to play the role of the IoT system. Each device is a wireless node of the considered network and certain specifications regarding node operation need to be taken into consideration in order to implement the previously described TiMAC policy. The particular information, along with various implementation issues (e.g., TDMA synchronization), is presented next. More specifically, the elements comprising the low-cost devices (nodes) are given in Section 4.1, and the types of messages being exchanged among the nodes are described in Section 4.2, TDMA parameters are presented in Section 4.3, and synchronization issues are given in Section 4.4.

4.1. Low-Cost Devices

As already mentioned, each device has to be low-cost and in this sense, industrial motes cannot be used because of their high cost. To achieve the low-cost goal, each node has been assembled using three essential elements. The first element is the core part of the node and is represented by an Arduino prototyping platform [18]. The rest of the elements are two RF modules operating at 433 MHz and are used for transmission and reception purposes. In Figure 2, one of the nodes comprising the system is shown. All three mentioned elements are installed, along with an LCD screen for experiment monitoring and debugging purposes.

Regarding the Arduino prototyping platform, it is an open-source and low-cost electronics platform, is easy to use, and capable of hosting various components (e.g., the RF modules considered in this paper). There are plenty of Arduino board designs utilizing a variety of microprocessors and controllers. The particular board used for this paper is the Arduino Uno Rev. 3. The latter is a board that is built on top of the ATMEL ATMEGA328 microcontroller with a clock speed of 16 MHz and 2 KB of SRAM [31].

As for the wireless communication part of the devices, it is handled by short-range RF modules operating in the license-free *Industrial, Scientific and Medical* (ISM) Band of 433.050–434.790 MHz. This particular band is free licensed to users for amateur and radio-location services [32]. Two different components are used to realize transmission (Tx) and reception (Rx), i.e., RWS-371-V1.0.3 [19] and TWS-BS-V1.0.3 [20], respectively. As already mentioned, these modules co-operate with Arduino and the wiring applied to them is schematically illustrated in Figure 3.

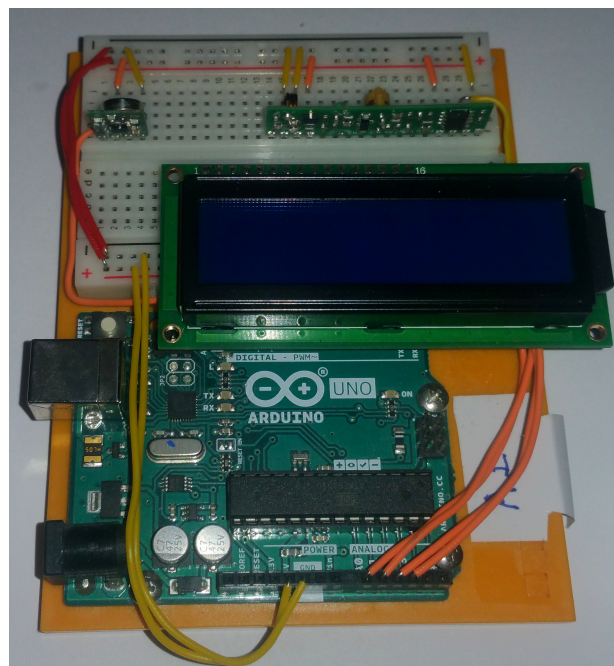


Figure 2. A device of the considered system enhanced with an LCD screen.

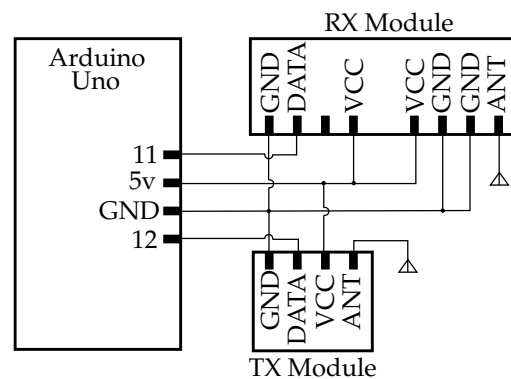


Figure 3. The wiring between the Arduino and the RF modules that implements the device for both transmission (i.e., Tx-module data connected to pin 12) and reception (i.e., Rx-module data connected to pin 12). The antenna (ANT) along with the power supply pins (i.e., GND and VCC) are also depicted.

4.2. Message Types

For the implementation of the TDMA protocol, two types of messages are exchanged among the network nodes of the considered low-cost IoT system. The group of messages that belong to the first type of messages are the SYNC messages. SYNC messages are transmitted by all network nodes at the beginning of each frame, for time synchronization purposes. The payload of SYNC messages “carries” the time remaining until the first time slot begins. The second type of messages contains the DATA messages. By the time a network node has data ready for transmission, it constructs a DATA message that contains the data to be sent. The next step is to transmit the DATA message 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. At the beginning of each frame, all nodes exchange and process SYNC messages in order to achieve synchronization of the receiver’s TDMA clock timer. DATA messages, carrying the valuable information nodes are supposed to send, are transmitted by the network nodes during the particular time-slots that have been assigned to them.

4.3. TDMA Parameters

On the one hand, low-cost devices and the elements comprising them ensure a low price for the final system. On the other hand, several limitations regarding frame timing parameters have to be taken into account when implementing the TDMA protocol on the selected ISM Band (of 433.050–434.790 MHz) using low-cost devices. More specifically, such limitations are: (i) nodes’ internal clock may deviate significantly [33]; (ii) the interference on the selected band may be high [34,35]; and (iii) the low-cost transmitter’s accuracy and receiver’s sensitivity may be reduced.

In order to tackle the effects and drawbacks of the above limitations, some countermeasures are taken. For avoiding collisions occurring due to the clock drifts, each time slot (t_{s_i}) of the TDMA frame is set to last for a considerably long period (i.e., 500 ms), whereas a 300 ms long guard time (t_g) is introduced between them for the same reason. For achieving the synchronization needed, at the beginning of each frame, nodes are allowed to get synchronized during a time period of 2000 ms (t_{st}). At the end of the frame, one more guard period, the frame guard period, is considered lasting 500 ms (t_{fg}), thus allowing for further node’s operations (e.g., sensing, message processing) to take place. The complete TDMA frame of the implemented policy is depicted in Figure 4.

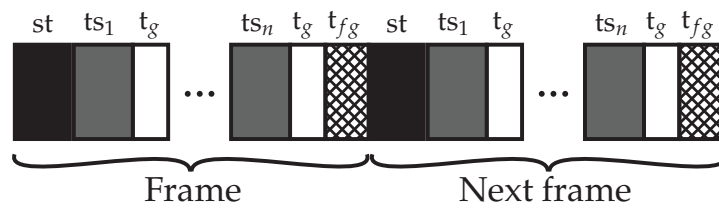


Figure 4. The TDMA frame of the implemented MAC protocol. Each time slot (ts_i) ends with the guard period (t_g). The synchronization time (st) along with the frame guard period (t_{fg}) are also depicted.

4.4. Synchronization

A significant factor in TDMA environments is the nodes’ awareness of the time that a frame begins. In order to fulfill the above requirement in multi-hop environments, a (multi-hop) decentralized clock synchronization mechanism has to be implemented. Timestamping techniques that have been extensively employed in the literature by several proposed synchronization algorithms (e.g., [36–38]) are employed here. Moreover, in the current work, the considered synchronization mechanism has to take into account that the focus is on the implementation of TiMAC on low-cost devices; thus, the results given consider the devices’ low-cost nature as well. In this sense, a long enough time period of 2000 ms is considered for nodes to synchronize their clocks. During this time period, starting at the beginning of each frame, nodes synchronize their clocks by transmitting the previously described SYNC messages. The latter process goes as follows: a reference node transmits a SYNC message to its neighbor nodes, thus informing them about the time remaining until the first time slot of the frame starts. 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.

Figure 5 depicts the considered synchronization mechanism between the selected reference node (r), and nodes u and v , connected in chain topology (i.e., $r-u-v$). In the example depicted, at the beginning of the synchronization period (st), the reference node r waits for a certain time interval t_{dr} and then transmits its clock $t_0 = st - t_{dr}$ to node u . When u receives the message, it 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 time interval (i.e., t_{du}) before transmitting its clock (i.e., t_1) to node v . Following the same procedure and by the time node v receives the message, it (i) changes its clock (i.e., $c_v = t_1 - t_{tx}$); (ii) waits for a certain amount of time (i.e., t_{dv}); and (iii) 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 will be ignored.

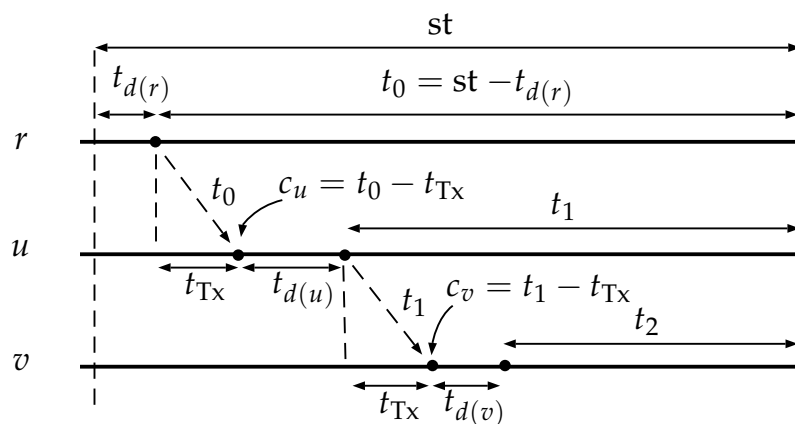


Figure 5. Illustration of the synchronization mechanism within a synchronization period (st), among nodes r , u and v connected in a chain topology. Nodes wait for a certain amount of time (i.e., t_{dr} , t_{du} and t_{dv} , respectively) before transmitting the time remaining until the first time slot (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.

5. Experimental Methodology and Results

A full graph topology is considered here for the experimental setup, consisting of five nodes. The considered network is assumed as part of a larger IoT network of maximum $N = 25$ nodes where the maximum number of neighbors is $D = 4$. Taking the above into consideration, the five nodes were set up in a full graph topology so as to simulate the maximum possible neighborhood of the assumed network. As a result, each node has exactly four neighbors, while collisions are expected to occur. The expectation regarding the performance of TiMAC is to accomplish the “at least one transmission per node per frame” goal, i.e., at least one transmission free of collisions. Note that transmissions may be unsuccessful for various reasons. For example, there might be interference in the 433 Mhz band, since the selected band is an ISM one and several devices placed near the system’s devices may also use it during the experiment. For avoiding as many such cases as possible, special care was taken to conduct the experiments in isolated areas (e.g., the basement of the premises of the University campus).

5.1. Methodology

As already mentioned, TiMAC is based on TDMA and, in this sense, its implementation is also based on TDMA. Considering the values deriving by the experimental system setup (i.e., $N = 25$ nodes and maximum number of neighbors $D = 4$), the corresponding values for k (i.e., $k = 1$) and q (i.e., $q = 5$) are extracted [17]. As a result, first-degree polynomials are going to be used for the slot allocation due to $k = 1$, and the length of the TDMA frame will be set at $q^2 = 25$, divided into five subframes of $q = 5$ slots each. Then, each node is randomly assigned a first-degree polynomial with coefficients from $GF(5)$ and, according to the latter polynomial, the particular time slot during which each node is allowed to transmit within each sub-frame is derived. Eventually, every node is allowed to transmit in five slots per frame.

Eleven different sets of polynomials are considered here, corresponding to the eleven different scenarios developed for the particular system. Consequently, different time-slots were assigned to each node per scenario and the obtained results are the summation of transmissions of 100 frames. Note that increased traffic conditions are assumed for the experimental setup, and therefore each node has always data ready for transmission during every time slot. As for the receivers of transmitted packets, for each node, one of its neighbors is arbitrarily selected as the destination of the transmitted packets. The latter selection of receivers remains unchanged throughout the experiments.

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

The main parameter under investigation is the *throughput* value each polynomial set (i.e., scenario) results to. 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.,

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

During the period of 100 frames, each node is eligible for transmission for up to 500 time-slots, whereas the number of total time-slots equals 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 at least one successful transmission per node per frame will take place. Consequently, the minimum network throughput is given by

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

5.2. Results

In Figure 6, the derived values of throughput for the eleven polynomial sets are depicted. It is observed here that, in all cases, throughput is higher than the minimum value guaranteed by the TiMAC policy (i.e., 0.04). In addition, it is clear that the first polynomial set (scenario) results in the highest value of throughput; however, it is smaller than the maximum value (i.e., 0.2). The latter result is expected since, on the one hand, the particular set of polynomials leads to a collision-free slot allocation. On the other hand, the channel interference, transmitter’s accuracy, and receiver’s sensitivity may provoke unsuccessful transmissions.

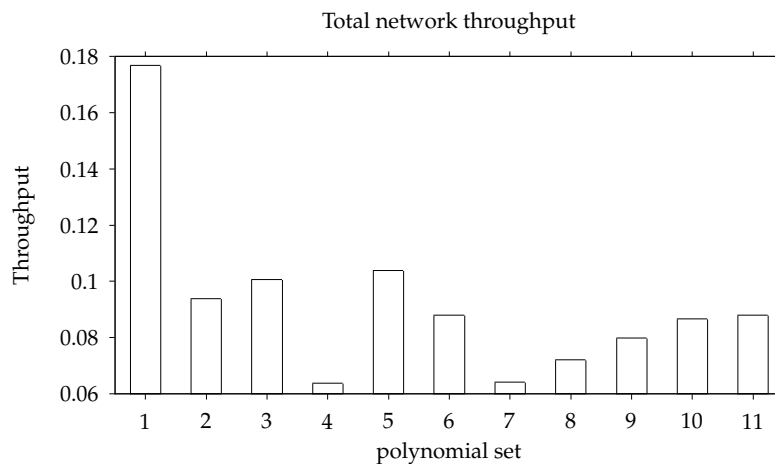


Figure 6. Throughput corresponding to various polynomial sets. In all cases, the throughput lays between the expected maximum (i.e., 0.2) and minimum (i.e., 0.04) values.

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, it is observed that the successful transmissions are significantly more than the successful transmissions in the other ten scenarios. However, they are less than the maximum theoretically expected value of five, which is the number of successful transmissions per node per frame in cases where no packets are lost for the duration of the 100 frames.

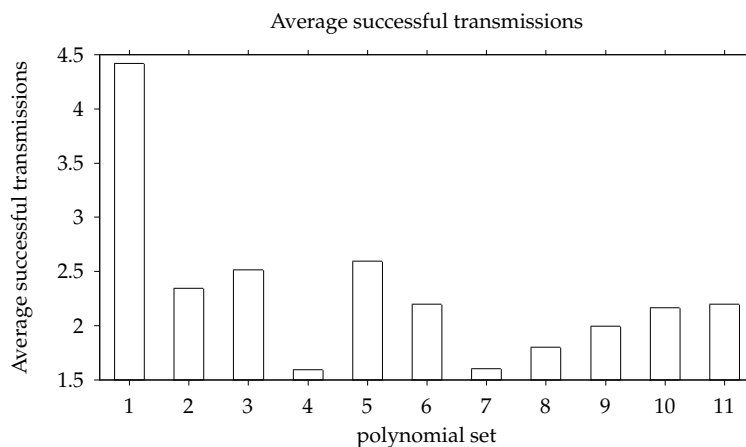


Figure 7. Average number of successful transmissions per frame per node corresponding to various polynomial sets. In all cases, the number of successful transmissions is between the expected maximum (i.e., 5) and minimum (i.e., 1) values.

The particular polynomials used in each set are listed in Table 1. Although the network topology remains unchanged throughout the experimental phase, each set leads to different number of successful transmissions.

The successful transmissions per polynomial set are listed in Table 2. Each set's sum of successful transmissions was used to derive the aforementioned throughput values.

Table 1. The particular polynomial sets used in the experiments.

Polynomial Set	1	2	3	4	5	6	7	8	9	10	11
Node A	$0x + 0$	$1x + 2$	$0x + 1$	$0x + 4$	$1x + 0$	$0x + 1$	$0x + 3$	$0x + 1$	$0x + 2$	$0x + 1$	$2x + 1$
Node B	$0x + 1$	$1x + 3$	$0x + 4$	$1x + 1$	$1x + 3$	$0x + 4$	$1x + 0$	$0x + 2$	$0x + 3$	$2x + 0$	$2x + 4$
Node C	$0x + 2$	$2x + 0$	$2x + 0$	$1x + 2$	$1x + 4$	$2x + 2$	$2x + 4$	$0x + 4$	$2x + 4$	$2x + 3$	$3x + 0$
Node D	$0x + 3$	$2x + 2$	$2x + 2$	$2x + 0$	$3x + 3$	$3x + 3$	$3x + 1$	$1x + 0$	$3x + 2$	$2x + 4$	$3x + 4$
Node E	$0x + 4$	$3x + 0$	$2x + 3$	$3x + 2$	$4x + 4$	$4x + 4$	$4x + 4$	$4x + 1$	$4x + 1$	$4x + 1$	$4x + 2$

Table 2. Successful transmissions per node for each polynomial set.

Polynomial Set	1	2	3	4	5	6	7	8	9	10	11
Node A	495	203	101	100	301	200	100	300	298	98	202
Node B	423	399	200	101	300	199	102	200	201	194	300
Node C	500	193	258	398	400	300	200	206	200	400	198
Node D	396	275	302	99	198	200	200	97	198	195	200
Node E	396	102	398	99	99	200	200	99	102	195	200
Total	2210	1172	1259	797	1298	1099	802	902	999	1082	1100

6. Conclusions

In this work, a MAC policy independent of the underlying topology (TiMAC) was implemented in a small IoT network consisting of low-cost devices based on Arduino and 433 MHz RF modules. A decentralized method was used for synchronizing the nodes' clocks as required from the underlying TDMA policy. Experiments were conducted in a full graph topology network under increased network traffic load.

The conducted experiments confirm in practice the theoretical expectations from TiMAC as all of the results were within the expected boundaries in terms of throughput and successful transmissions per frame. These results suggest that policies that are independent of the underlying network topology could be considered when designing an IoT system, as they offer a means of node deployment without the usual considerations of traditional TDMA schemes (physical location combined with particular slot assignment), which could prove to be easier due to the nature of this type of networks and still could meet the strict time constraints of various delay-sensitive applications.

The decentralized synchronization method used here also proved to be successful. At the same time, the fact that a particular experimental setup was close to the theoretical maximum expected values suggests that the proposed policy can be improved if unique characteristics of particular polynomial sets are taken into account before assigning them to network nodes.

Future work should experiment with larger sensor networks so that TiMAC may be tested in multi-hop environment. Moreover, cybersecurity and resilience will have to be taken into account, as both of them are crucial parts of IoT, while their performance on low-cost devices is questionable.

Author Contributions: Conceptualization, G.T., V.D. and A.P.; methodology, V.D. and G.T.; validation, G.K., A.P. and K.O.; formal analysis, G.K., K.O. and C.T.A.; investigation, G.K., V.D. and A.P.; resources, K.O. and C.T.A.; writing—original draft preparation, G.T.; writing—review and editing, K.O. and C.T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: 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 and National Greek Funds (ESPA) under the Regional Operational Program “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 Program 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.). In addition, this research is co-financed by Greece and the European Union (European Social Fund—ESF) through the Operational Program “Human Resources Development, Education and Lifelong Learning” in the context of the project “Reinforcement of Postdoctoral Researchers—2nd Cycle” (MIS-5033021), implemented by the State Scholarships Foundation (IKY).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MAC	Medium Access Control
TDMA	Time Division Multiple Access
IoT	Internet of Things
TiMAC	Topology independent Medium Access Control
CSMA	Carrier Sense Multiple Access

References

- Osseiran, A.; Monserrat, J.F.; Marsch, P. *5G Mobile and Wireless Communications Technology*; Cambridge University Press: Cambridge, UK, 2016.
- Andrews, J.G.; Buzzi, S.; Choi, W.; Hanly, S.V.; Lozano, A.; Soong, A.C.; Zhang, J.C. What will 5G be? *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1065–1082. [[CrossRef](#)]
- Ashton, K. That ‘internet of things’ thing. *RFID J.* **2009**, *22*, 97–114.
- Ni, J.; Zhang, K.; Lin, X.; Shen, X.S. Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 601–628. [[CrossRef](#)]
- GSMA Intelligence. Definitive Data and Analysis for the Mobile Industry. *GSMA Intell.* **2016**.
- Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [[CrossRef](#)]
- Vaquero, L.M.; Rodero-Merino, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 27–32. [[CrossRef](#)]
- Xiong, X.; Zheng, K.; Xu, R.; Xiang, W.; Chatzimisios, P. Low power wide area machine-to-machine networks: Key techniques and prototype. *IEEE Commun. Mag.* **2015**, *53*, 64–71. [[CrossRef](#)]
- Suriyachai, P.; Roedig, U.; Scott, A. A survey of MAC protocols for mission-critical applications in wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 240–264. [[CrossRef](#)]
- Sherazi, H.H.R.; Grieco, L.A.; Boggia, G. A comprehensive review on energy harvesting MAC protocols in WSNs: Challenges and tradeoffs. *Ad Hoc Netw.* **2018**, *71*, 117–134. [[CrossRef](#)]
- Dragonas, V.; Oikonomou, K.; Giannakis, K.; Stavrakakis, I. A Disjoint Frame Topology-Independent TDMA MAC Policy for Safety Applications in Vehicular Networks. *Ad Hoc Netw.* **2018**, *79*, 43–52. [[CrossRef](#)]
- Hadded, M.; Mühlethaler, P.; Laouiti, A.; Zagrouba, R.; Saïdane, L.A. TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2461–2492. [[CrossRef](#)]
- Ergen, S.C.; Varaiya, P. TDMA Scheduling Algorithms for Wireless Sensor Networks. *Wirel. Netw.* **2010**, *16*, 985–997. [[CrossRef](#)]
- Wang, F.; Zeng, P.; Yu, H. Slot Time Synchronization for TDMA-Based Ad Hoc Networks. In Proceedings of the 2008 International Symposium on Computer Science and Computational Technology, Shanghai, China, 20–22 December 2008; Volume 2, pp. 544–548. [[CrossRef](#)]
- Sundaraman, B.; Buy, U.; Kshemkalyani, A.D. Clock synchronization for wireless sensor networks: A Survey. *Ad Hoc Netw.* **2005**, *3*, 281–323. [[CrossRef](#)]

16. Rhee, I.K.; Lee, J.; Kim, J.; Serpedin, E.; Wu, Y.C. Clock Synchronization in Wireless Sensor Networks: An Overview. *Sensors* **2009**, *9*, 56–85. [CrossRef]
17. Chlamtac, I.; Faragó, A. Making transmission schedules immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Trans. Netw.* **1994**, *2*, 23–29. [CrossRef]
18. Banzi, M.; Shiloh, M. *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*; Maker Media, Inc.: San Francisco, CA, USA, 2014.
19. WENSHING ELECTRONICS CO., LTD. *Wireless Hi Power Transmitter Module (RF ASK)*; v1.0.3; WENSHING ELECTRONICS CO., LTD.: Taipei, Taiwan, 2010.
20. WENSHING ELECTRONICS CO., LTD. *Wireless Hi Sensitivity Receiver Module (RF ASK)*; v1.0.3; WENSHING ELECTRONICS CO., LTD.: Taipei, Taiwan, 2008.
21. Oliveira, L.; Rodrigues, J.J.; Kozlov, S.A.; Rabêlo, R.A.; Albuquerque, V.H.C.d. MAC layer protocols for internet of things: A survey. *Future Internet* **2019**, *11*, 16. [CrossRef]
22. Kumar, A.; Zhao, M.; Wong, K.J.; Guan, Y.L.; Chong, P.H.J. A comprehensive study of iot and wsn mac protocols: Research issues, challenges and opportunities. *IEEE Access* **2018**, *6*, 76228–76262. [CrossRef]
23. Ferrer, T.; Céspedes, S.; Becerra, A. Review and evaluation of MAC protocols for satellite IoT systems using nanosatellites. *Sensors* **2019**, *19*, 1947. [CrossRef]
24. Ye, Q.; Zhuang, W. Distributed and adaptive medium access control for Internet-of-Things-enabled mobile networks. *IEEE Internet Things J.* **2016**, *4*, 446–460. [CrossRef]
25. Ye, Q.; Zhuang, W. Token-based adaptive MAC for a two-hop Internet-of-Things enabled MANET. *IEEE Internet Things J.* **2017**, *4*, 1739–1753. [CrossRef]
26. Shahin, N.; Ali, R.; Kim, Y.T. Hybrid slotted-csma/ca-tdma for efficient massive registration of iot devices. *IEEE Access* **2018**, *6*, 18366–18382. [CrossRef]
27. Kim, Y.J.; Choi, H.H.; Lee, J.R. A bioinspired fair resource-allocation algorithm for TDMA-based distributed sensor networks for IoT. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 7296359. [CrossRef]
28. Li, Y.; Zhang, X.; Zeng, J.; Wan, Y.; Ma, F. A distributed TDMA scheduling algorithm based on energy-topology factor in Internet of Things. *IEEE Access* **2017**, *5*, 10757–10768. [CrossRef]
29. Li, Y.; Zhang, X.; Qiu, T.; Zeng, J.; Hu, P. A Distributed TDMA Scheduling Algorithm Based on Exponential Backoff Rule and Energy-Topology Factor in Internet of Things. *IEEE Access* **2017**, *5*, 20866–20879. [CrossRef]
30. Stewart, I.N. *Galois Theory*; CRC Press: Boca Raton, FL, USA, 2015.
31. Arduino LLC. *Arduino*; Rev3; Arduino LLC: Boston, MA, USA, 2010.
32. International Telecommunication Union (ITU). Radio Regulations. 2016. Available online: <https://www.itu.int/pub/R-REG-RR-2016> (accessed on 15 April 2020).
33. Guo, C.; Shen, J.; Sun, Y.; Ying, N. RB Particle Filter Time Synchronization Algorithm Based on the DPM Model. *Sensors* **2015**, *15*, 22249–22265. [CrossRef]
34. *IEEE Recommended Practice for Powering and Grounding Electronic Equipment*; IEEE Std 1100-2005 (Revision of IEEE Std 1100-1999); IEEE: Piscataway, NJ, USA, 2006; pp. 1–703. [CrossRef]
35. Robinson, D.; Wysocki, T.A.; Smith, V.W.; Popovski, K. *Background Radio Frequency Interference Measurements for Wireless Devices in the Electricity Supply Industry*; University of Nebraska: Lincoln, NE, USA, 2005.
36. Maróti, M.; Kusy, B.; Simon, G.; Lédeczi, Á. The flooding time synchronization protocol. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 39–49.
37. Ganeriwal, S.; Kumar, R.; Srivastava, M.B. Timing-sync protocol for sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 138–149.
38. Skiadopoulos, K.; Tsipis, A.; Giannakis, K.; Koufoudakis, G.; Christopoulou, E.; Oikonomou, K.; Kormentzas, G.; Stavarakakis, I. Synchronization of Data Measurements in Wireless Sensor Networks for IoT Applications. *Ad Hoc Netw.* **2019**, *89*, 47–57. [CrossRef]

