

Research Article

An Application-Aware Clustering Protocol for Wireless Sensor Networks to Provide QoS Management

Alba Rozas  and Alvaro Araujo 

B105 Electronic Systems Lab, Department of Electronic Engineering, Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT), Universidad Politécnica de Madrid (UPM), Av. Complutense 30, 28040 Madrid, Spain

Correspondence should be addressed to Alba Rozas; albarc@b105.upm.es

Received 14 May 2019; Accepted 29 July 2019; Published 2 September 2019

Academic Editor: Carlos Ruiz

Copyright © 2019 Alba Rozas and Alvaro Araujo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) have long been established as a suitable technology for gathering and processing information from the environment. However, recent applications and new multimedia sensors have increased the demand for a more adequate management of their quality of service (QoS). The constraints and demands for this QoS management greatly depend on each individual network's purpose or application. Low-Energy Adaptive Clustering Hierarchy (LEACH) is arguably the most well-known routing protocol for WSNs, but it is not QoS-aware. In this paper, we propose LEACH-APP, a new clustering protocol based on LEACH that takes the network's application into account and is aimed at providing a better overall QoS management. We thoroughly describe our proposal and provide a case study to explain its operation. Then, we evaluate its performance in terms of two significant QoS metrics—throughput and latency—and compare it to that of the original protocol. Our experiments show that LEACH-APP increases the throughput by roughly 250% and reduces the latency by almost 80%, overall providing a more flexible and powerful QoS management.

1. Introduction

The Internet of Things (IoT) has recently emerged with new applications and services to connect and give access to all kinds of devices from the physical world. Wireless sensor networks (WSNs) are one of the key infrastructures that provide support to the IoT paradigm. They consist of a set of small electronic devices, commonly called nodes, which share and collect information about their surroundings. As the report in [1] points out, Machine To Machine (M2M) connections, which include the IoT and WSN, will grow worldwide from 6.1 billion in 2017 to 14.6 billion by 2022, largely surpassing the global population.

These technologies are deployed in numerous scenarios ranging from smart cities to health applications or the monitoring of critical infrastructures. These are demanding environments that usually require the networks to have a long lifespan and need little to no maintenance. Since nodes are usually powered from a limited battery, this means that energy efficiency is one of the most critical challenges in these

kinds of networks. Also, the appearance of new multimedia sensors and data-rich applications has raised the need for a careful consideration and management of their quality of service (QoS).

Within the topic of energy efficiency, a very common approach by researchers focuses on optimizing the transmission of information throughout the network, creating efficient routing protocols. In the field of WSNs, the hierarchical protocol Low-Energy Adaptive Clustering Hierarchy (LEACH) [2] is arguably the most used and researched. LEACH has been proved to effectively reduce energy consumption by grouping the network nodes into clusters, aggregating their data, and consequently reducing the number of radio transmissions. Since its publication, a lot of extensions and modifications have been developed by researchers using this protocol as the basis [3].

LEACH and all its extensions and modifications approach the issue of energy efficiency regardless of the network's particular application or purpose. However, different WSN scenarios have different requirements and can

consequently benefit from different optimizations adapted to their specific characteristics. This is particularly true in terms of QoS metrics such as the throughput, the latency, or the jitter. The authors in [4] demonstrate that taking into account the network's application can effectively increase its lifetime while also providing a better QoS management and a better control of its degradation process.

In this paper, we propose a new energy-efficient protocol based on LEACH that considers the network's purpose and provides an enhanced performance in terms of QoS. This modification consists in separating the network's traffic into different categories according to their importance for the application. These different traffic streams are assigned more or less resources from the cluster head according to their priority level. Using a well-known WSN simulator, we test our algorithm in various realistic scenarios under different conditions. The results show that our proposed protocol has a better performance than original LEACH in terms of two QoS metrics: throughput and latency.

The rest of this paper is organized as follows: Section 2 reviews some relevant works in this field. Section 3 thoroughly describes the proposed protocol providing a motivation for its development and a case study. In Section 4, we evaluate its performance in terms of QoS and compare it with that of the original protocol. Finally, in Section 5, we present some conclusions derived from this research.

2. Related Works

Since nodes usually have a limited power supply, energy efficiency is one of the key challenges in WSNs. Several research works have stated that radio communication is the most expensive operation in terms of energy in these kinds of networks [5, 6]. Also, this consumption is directly proportional to the distance between the source and destination nodes. The communication module in WSN nodes usually consists of a radio transceiver, some RF circuitry, and an antenna. Radio transceivers generally have a sleep mode in which power consumption is reduced by at least an order of magnitude in relation to their transmission, reception, or idle state. Thus, minimizing the number of transmissions and putting the transceiver in a sleep mode whenever possible achieve a significant reduction in energy consumption [7].

Hierarchical routing protocols are based on grouping the sensor nodes into several clusters according to their physical location. Sensor nodes send their data to a special node in the cluster, usually called the cluster head (CH). The cluster head aggregates the received data and forwards them to the base station (BS). It can also do some processing to remove redundant correlated readings from nearby sensors. As a result, clustering protocols reduce the number of radio transmissions and their distances, thus decreasing the global energy consumption [8, 9].

LEACH [2] is the most popular clustering routing protocol for WSNs focused on reducing energy consumption. Its key idea consists in performing a Time Division Multiple Access (TDMA) scheduling for the intracluster communication. This way, sensor nodes send their data to the cluster head in their allotted time slot and spend the rest of the cycle

in a sleep mode. In each round, the CH is selected probabilistically and, in a distributed way, without the need for the nodes to exchange control messages. Also, the cluster head duties rotate among all the nodes so that their energy is spent uniformly. This way, LEACH achieves an energy reduction of a factor between 7x and 8x compared to direct communication between the nodes and the BS [9].

However, despite these advantages, there are also a few drawbacks to the original LEACH protocol. Firstly, the CH is selected randomly and the probability of a node to become the CH is the same for all of them, regardless of their remaining energy. This can greatly affect the network lifetime and its degradation. Also, this random selection does not take into account the node position and network topology, so clusters may be distributed unequally.

Since its publication in 2000, several modifications and extensions have been made to the original LEACH protocol. These modifications and improvements have been done to address different aspects and challenges of LEACH, such as the ones abovementioned, and the WSN paradigm in general. The authors in [3] classify the LEACH variants into eight categories according to their objective: energy efficiency, load balancing, coverage and connectivity, scalability, data fusion, minimum delay, security, and robustness.

Very few of these LEACH variations focus on the topic of QoS. In fact, the QoS has not been given great importance within the WSN research field until recent years. Traditionally, WSN energy and processing constraints made applications with high data rates impossible. WSN applications have almost always been aimed at sensing and monitoring certain variables with low measuring frequency and very light data.

Oppositely, the improvement of WSN capabilities and the emergence of new multimedia and image-based sensors have allowed new applications to appear. These new applications have much higher data rates and manage more complex data structures. Consequently, their QoS has to be very carefully monitored and maintained.

The work in [10] examines QoS aspects in the field of WSN clustering protocols. Its authors propose a new protocol for image-based applications and measure its performance in terms of the end-to-end delay, a very well-known QoS metric. The authors in [11] also propose a new WSN clustering protocol that improves QoS metrics such as the average delay, packet loss ratio, and throughput, when compared with a static sink scenario. Another protocol that uses an ad hoc cluster-based architecture adapted to the network topology is proposed in [12]. Its clusters are created and organized according to QoS parameters such as the bandwidth, delay, or packet loss, providing an end-to-end QoS for each data stream. Recently, the authors in [13] propose a QoS-aware clustering routing protocol for Wireless Sensor and Actuator Networks (WSANs). It provides QoS in terms of delay, but it is particularly adapted to the heterogeneous characteristics of WSANs and not directly applicable to WSNs. None of these works are based on or related to LEACH.

In turn, the authors in [14] propose a modified LEACH algorithm where the cluster head selection algorithm takes

into account the nodes' residual energy. They use QoS parameters such as the throughput, the packet delivery ratio, or the delay to evaluate the protocol's performance against original LEACH. However, the algorithm's main objective is to reduce energy consumption and improve the network lifetime, not to improve specific QoS aspects. The same occurs with the work in [15], whose authors introduce a multilevel LEACH variant that reduces consumption, extends the lifetime, and increases the throughput of the network. These works only use QoS metrics to evaluate performance but not as an active feature of their algorithms.

To the best of our knowledge, there are no energy-efficient clustering protocols for WSNs that take into account the network's specific purpose or application to manage its QoS. Particularly, there are no QoS-oriented LEACH modifications other than the ones mentioned. In this paper, we present LEACH-APP, a new clustering protocol based on LEACH which takes the network application into consideration and is specifically aimed at improving its QoS.

3. LEACH-APP: The Proposed Protocol

3.1. Motivation. A lot of WSN applications would benefit from a careful monitoring and maintenance of their QoS. The QoS is intrinsically related to the network's purpose, so different WSN scenarios have different QoS requirements. This can be seen considering a smart home monitoring system with several services or features based on a WSN. An intrusion surveillance service may have a strict upper bound on its latency when triggering an alarm. On the other hand, a service for monitoring and adjusting room temperature should not be particularly demanding in terms of delay or latency.

In addition, these QoS requirements could change over time in response to changes in the network's circumstances. When the network nodes are low on battery, they may not be able to keep providing the same QoS for all their traffic as in the beginning of their operation. In the previous example, when energy starts depleting, the temperature service could be downsized or completely disabled. This way, its liberated resources can be used to keep providing an adequate QoS for the intrusion detection system. This sort of actions and approaches is collectively called controlling the network's degradation by the authors in [4]. By doing this, the QoS can be monitored and at least partially maintained for as long as the network is considered functional.

Having analyzed these characteristics, we have developed a new protocol based on LEACH that considers the possibility of a network with different types of traffic. These different traffic streams are grouped into several categories according to their importance for the network's application, establishing several priority levels. This prioritization is dynamic and can be modified over time to attend to changes in the environment or inputs from the user.

3.2. Protocol Operation. As is the case with original LEACH, our protocol's operation consists of several rounds, each of them divided into two phases: the set-up and the steady phases. In the set-up phase, nodes individually determine if

they are going to be the cluster head for that particular round based on a probabilistic algorithm. If they are, they advertise that information to the rest of the network. Non-CH nodes receive these advertisement messages, decide which cluster to join based on proximity, and inform the corresponding CH. After that, the CH creates a TDMA schedule for all the nodes in its cluster and broadcasts it back to them. The operation of the set-up phase in LEACH-APP is very similar to that of original LEACH. The only difference introduced by LEACH-APP is that the JOIN request packet sent from nodes to CHs has an extra field indicating the priority level of the requesting node. This allows the CH to determine how many extra slots—if any—should be given to each particular node in the TDMA schedule.

At this point, the steady phase begins. Nodes transmit their packets only in their allotted slot of the TDMA schedule and stay in a sleep state the rest of the time. The flowchart in Figure 1 illustrates this operation and shows the decisions and actions taken by the nodes, according to their roles. It also displays the messages exchanged in the set-up and steady phases.

The TDMA schedule simply consists of all cluster members ordered by the time they joined the cluster. Each iteration of this schedule from beginning to end is called a frame. There are several frames in a round, depending on the slot length and the round length, two of the most important parameters. Since these parameters are independent from each other and the number of members of a cluster is not known a priori, the amount of frames per round may not be a whole number. The different phases and timing structure of LEACH and LEACH-APP can be seen in Figure 2.

The round-based operation, the CH probabilistic selection, and the TDMA intracluster communication are all characteristics of original LEACH. Our proposed protocol maintains these characteristics and introduces a mechanism for managing and optimizing the QoS by modifying the TDMA scheduling scheme. The proposed modification barely affects the original mechanism, hardly adding any overhead to the original operation. Particularly, we add a new field in the JOIN request message, stating the priority level (PL) of the requesting node. This priority level is taken into account by the CH in the creation of the TDMA schedule, adding extra time slots for the higher priority nodes. The priority level is set according to the network application: the lower the priority level number, the higher the node's importance to the application. The percentage of added extra slots, ESP1 and ESP2, are parameters dynamically modifiable by the application. The different priority levels and their effect can be seen in Table 1.

The introduction of priority level 0 is arguably the most novel aspect of our proposal since it introduces a way for nodes to send packets regardless of LEACH's TDMA schedule. This way, high-priority traffic can be better served by the network and its QoS significantly improved. A possible drawback of this provision is that a PL0 node with a high data rate could cause packet collisions with neighboring nodes, inside or outside its cluster. The case of intercluster collisions is already addressed by the original LEACH protocol by using

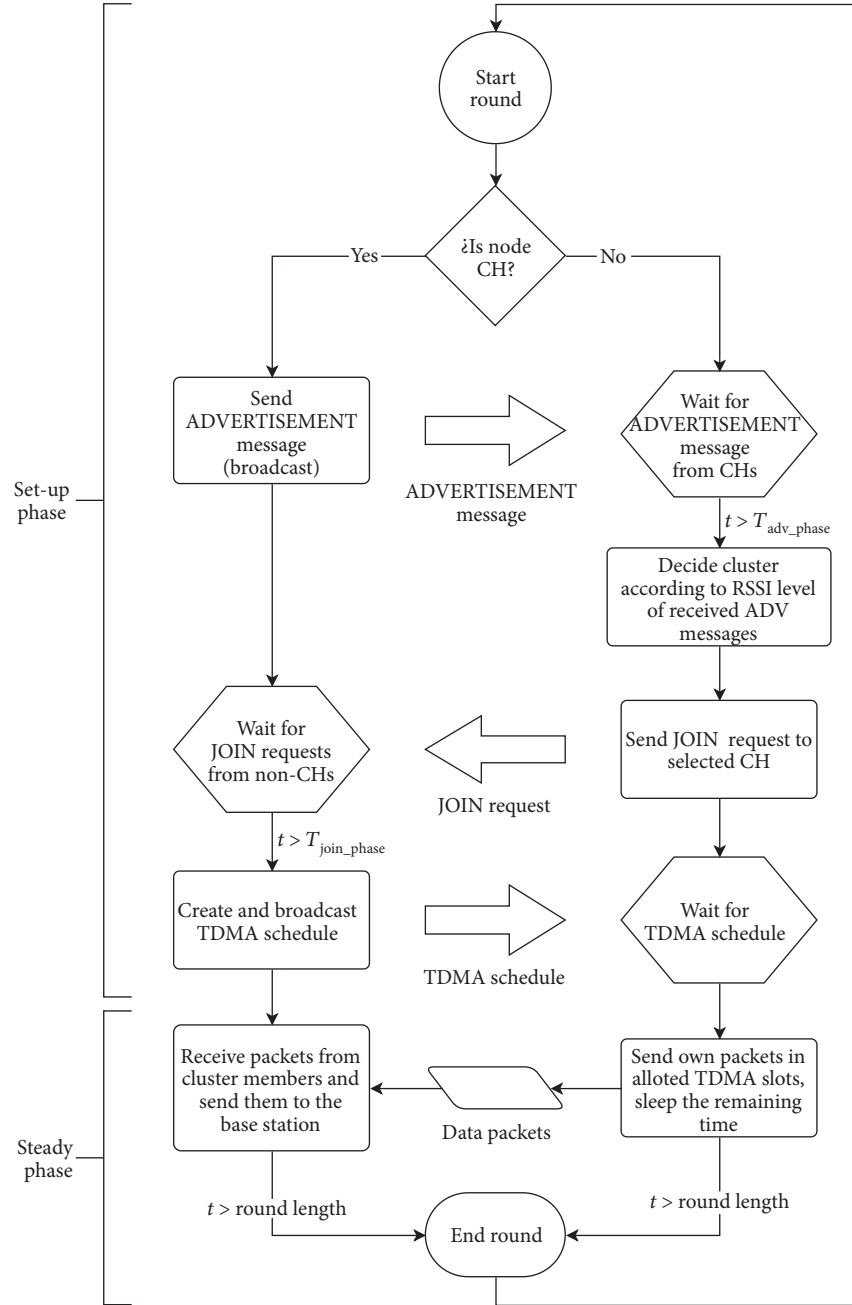


FIGURE 1: Flowchart illustrating the set-up phase of LEACH-APP.

a CDMA scheme with different codes for nearby clusters. In the case of intracluster collisions caused by PL0 traffic, it would be advised to add collision avoidance mechanisms, such as CSMA, to operate alongside our algorithm in lower layers of the protocol stack.

Our protocol provides a mechanism for the application to manage the QoS of its different traffic streams according to their importance. The application developer has to categorize these traffic streams and assign them their corresponding priority levels. How the application performs this prioritization, or the existence of a methodology for this categorization, is the responsibility of the developer.

3.3. Case Study. To further illustrate our proposal, we consider the case study of a cluster with ten members in a scenario with the parameters presented in Table 2. These parameters were selected with the purpose of having an illustrative case study where the operation of our protocol was well explained, and the number of added extra slots was easily obtained and represented. It is important to note that LEACH-APP does not set these parameters, and their selection is a responsibility of the application developer.

The ten cluster members are numbered from 1 to 10, each of them with the priority levels shown in Table 3.

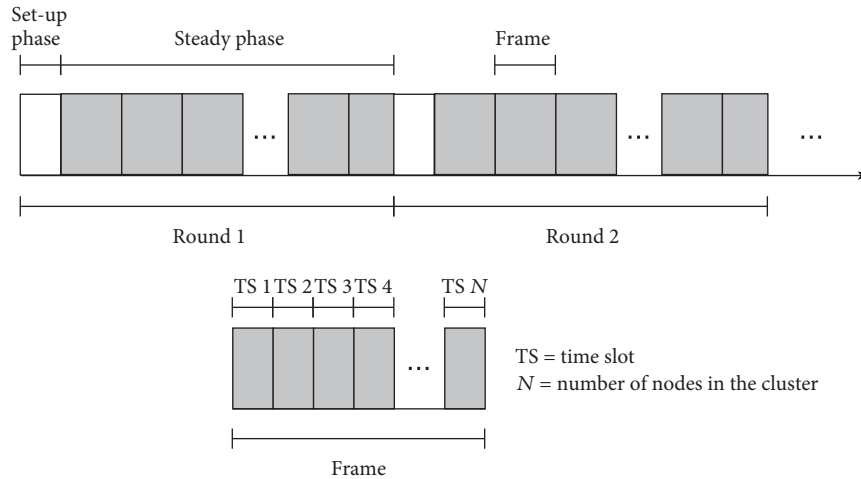


FIGURE 2: LEACH and LEACH-APP intracluster timing structure.

TABLE 1: Priority levels and their effect in the TDMA schedule in LEACH-APP.

Priority level	Effect
0	The node does not have to follow the TDMA schedule ¹
1	ESP1 % extra slots added with respect to the number of cluster members
2	ESP2 % extra slots added with respect to the number of cluster members
3	No extra slots are added to the TDMA schedule

¹It can send packets at any time, and the CH forwards them to the sink immediately.

TABLE 2: LEACH-APP parameters in the case study.

Parameter	Value
Set-up phase length	2 s
Slot length	0.2 s
Round length	20 s
ESP1, percentage of extra slots for PL1 nodes	30%
ESP2, percentage of extra slots for PL2 nodes	10%

TABLE 3: Node priority levels in the case study.

Nodes	Priority level
1	0
2 and 3	1
4 and 5	2
6, 7, 8, 9, and 10	3

In this situation, the CH should ignore node 1 in the schedule since it has priority level 0. Also, it should add 30% extra slots with respect to the number cluster members

to nodes 2 and 3. Since there are 10 members, 3 extra slots should be added to each of these nodes. In turn, nodes with priority level 2 get one extra slot each. The original LEACH frame for this scenario can be seen in Figure 3, along with the one for the proposed LEACH-APP protocol.

The extra slots are allocated at the beginning of the schedule because of the repetitive nature of frames over the round and the fact that the frames per round ratio does not have to be a whole number. This means that at the end of the round, the last frame would likely not have time to finish completely. Having the extra slots, for the highest priority traffic, at the beginning of the frame increases their occurrence rate in the whole round.

The effect of our algorithm can be seen by calculating the percentage of the total round length allotted to each type of traffic. Since the round length is 20 s and the set-up phase takes 2 s, the steady phase has a fixed length of 18 s. This means that in original LEACH the frame could be repeated

$$\frac{18 \text{ s}}{2 \text{ s}} = 9 \text{ times.} \quad (1)$$

On the other hand, in LEACH-APP, the frame can be repeated

$$\frac{18 \text{ s}}{3.4 \text{ s}} = 5.29 \text{ times,} \quad (2)$$

in the round. This consists of 5 complete frames and a final one with only the first 5 slots.

We consider now the particular case of node 2 with a priority level of 1. In original LEACH, it had allocated one slot per frame, i.e., 9 slots over the course of the round. This means

$$9 \times 0.2 = 1.8 \text{ s,} \quad (3)$$

allotted out of every 20 s, i.e., 9% of the total time. Conversely, this same node in LEACH-APP gets 4 slots per frame, i.e.,

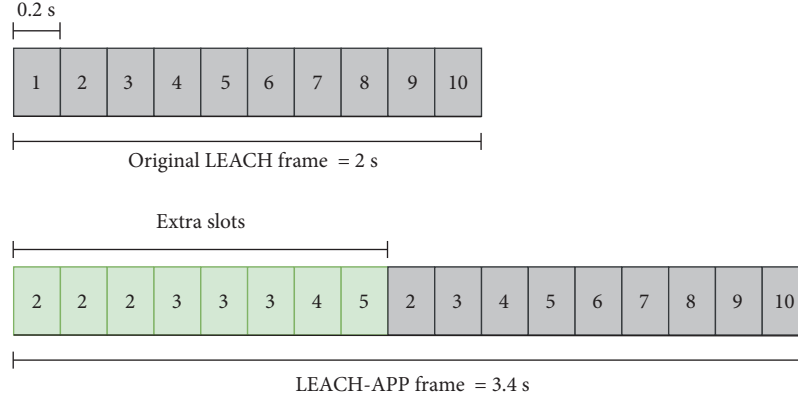


FIGURE 3: TDMA frames of the case study in original LEACH and LEACH-APP.

20 slots in the 5 complete frames and three more slots in the final frame, for a total of 23 slots. This means

$$23 \times 0.2 = 4.6 \text{ s}, \quad (4)$$

out of every 20 s, 23% of the total round time. In the following section, we will present proof that this increase in the allocated time achieves a significant improvement on several QoS metrics.

Apart from its positive impact on QoS, our proposed algorithm has several other advantages. Firstly, given its dynamic nature, it allows the user to modify his or her prioritization scheme at any particular time. These changes are rapidly adopted, taking effect at the beginning of the following round. Also, the overhead introduced by our algorithm is almost negligible, consisting only of a field of two bits in the JOIN request message to codify the priority level and some extra processing at the creation of the TDMA schedule by the CHs.

In turn, the extra time allotted to priority traffic in our algorithm has the consequence of a collective decrease of resources allocated to nonpriority traffic. In this case study, nodes with a priority level of 3 have an allocated time of 9% of the total round length in original LEACH. However, with LEACH-APP, they are only allotted 1 slot per frame, i.e., 5 slots in the complete round. This consists of

$$5 \times 0.2 = 1 \text{ s}, \quad (5)$$

out of every 20 s, 5% of the total time.

This apparent drawback is the consequence of taking into account the particular characteristics of the network traffic at any given moment, applying a prioritization scheme aligned with the user preferences. This is justifiable in applications with different types of traffic and provides an arguably better QoS management than treating all information in the same way.

3.4. Application Examples. Several WSN scenarios would benefit from managing the QoS of their traffic with our protocol. Generally, any network that harbors different applications or applications with multiple services is a

candidate for applying our traffic categorization and prioritization scheme.

An example would be a hospital or nursing home with a WSN to monitor a patient's vital signs. In a medical context, different biological variables have very different requirements regarding the frequency of their acquisition, the amount of data to transmit, or the tolerance for missing or delayed information. For instance, a wireless ECG application requires a high throughput and very low latency and explicitly needs the information from all its electrodes. Oppositely, a body temperature monitoring service requires a much lower throughput and has a higher tolerance for delays and packet loss. Using our proposed protocol, the ECG traffic would have a higher priority level than the one of the body temperature service, and their QoS would be managed accordingly.

Another example is a smart home scenario with a multi-service monitoring application. One of its possible features could be a light-sensing service that manages blinds, curtains, and artificial light in response to the natural light that reaches a given room. On the other hand, the same network may include a surveillance and intrusion detection system based on cameras and image sensors. It is easy to see that the QoS requirements for these two services are radically different and that the network would benefit from applying our prioritization scheme. This would be even more important in the performance degradation process that occurs at the end of the network lifetime when nodes start to deplete their batteries and some traffic cannot be properly handled.

4. Performance Evaluation

To evaluate the performance of our proposed algorithm LEACH-APP, we have simulated its behavior in several scenarios using Castalia [16]. Castalia is a well-known WSN simulator based on OMNeT++, whose last version was released in 2013. Its main feature is the realistic nature in which it simulates the radio channel, providing a very accurate scenario in terms of signal loss, packet collision, and interference.

We have compared its performance with that of original LEACH in terms of two QoS metrics: the throughput of packets received at the sink and the latency of those packets

TABLE 4: Baseline scenario parameters.

Simulation parameter	Value
Simulation run time	200 s
Network size	70 × 70 m
Number of nodes	100
Node distribution	10 × 10 grid
Node model	Texas Instruments CC2420
Initial energy	18720 J
Transmission power	0 dBm
Packet size	20 bytes
Transmission energy cost per packet	93.59 μ J
Reception mode power consumption	62 mW
Packet rate	10 packets/s
Percentage of CHs	5%
Round length	20 s
Slot length	0.2 s
ESP1	30%
ESP2	10%

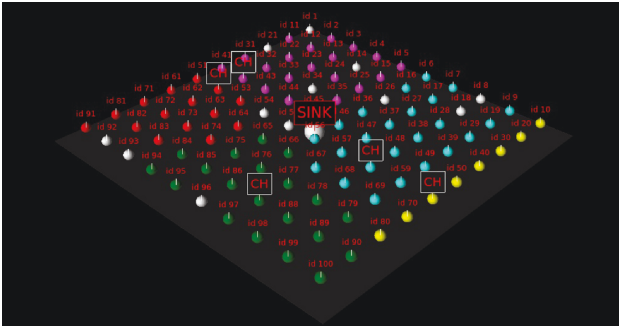


FIGURE 4: Network topology with colors representing the different clusters in a given round.

from source to sink. In order to perform these simulations, we have fully implemented LEACH-APP in Castalia, based on an implementation of the original LEACH protocol made by the authors in [17].

4.1. Simulation Set-Up. The network parameters used in our simulations are listed in Table 4. There are several processes in Castalia with a random nature: shadowing in the wireless channel, interference, random decisions at the MAC layer, etc. In order to smooth the variations produced by this randomness and have a clearer vision of the impact of our algorithm, we have run 5 repetitions of each unique set-up. The basic scenario consists of a square of 70 × 70 m, with 100 regular nodes distributed in a 10 × 10 grid. There is also a sink occupying a spot next to the center of the square, as seen in Figure 4. Castalia calculates the free space path loss using the lognormal shadowing model [16]. The transmission energy cost for each sent packet is calculated multiplying the power consumption in a transmission mode—57.42 mW—by the time it takes to send a packet of the specified length, as

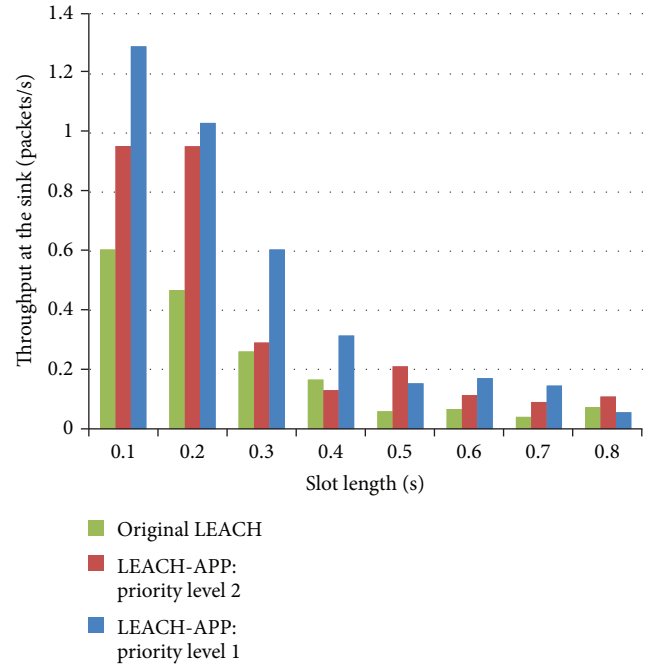


FIGURE 5: LEACH-APP vs. original LEACH: throughput of packets at the sink.

referred in [18]. All receiving nodes—CHs and sink—spend all their time in a reception mode, with the power consumption specified in the table. The rest of the parameters have typical values for this field. The bottom five parameters are specific for LEACH and/or LEACH-APP. They will vary from the ones in this table in certain experiments, when their particular impact is being evaluated. The values from Table 4 should be assumed for all the nodes in the network unless there is an explicit statement saying otherwise.

4.2. Results and Discussion

4.2.1. LEACH-APP vs. Original LEACH: Performance Comparison. In the first experiment, we evaluate the effect of giving a higher priority level and subsequently assigning extra slots to one of the nodes in the network (node 43 in particular, but it could be any other). We have simulated the baseline scenario under three different configurations: with original LEACH, with LEACH-APP and giving the node a priority level of 1, and with LEACH-APP and giving the node a priority level of 2. When simulating LEACH-APP, the remaining 99 nodes have a default priority level of 3; i.e. no extra slots are allotted, so their traffic is handled in the same way as in original LEACH.

Figure 5 shows the throughput at the sink of packets sourced from node 43 in the three cases, for different values of the slot length. The packet rate of this node is 80 packets per second. It is worth noting that due to its realistic modeling of the wireless channel, the packet loss rate is quite high in all Castalia simulations, regardless of the use of LEACH, LEACH-APP, or any other protocol [19].

From the graph above, we can see that the throughput of a particular node is effectively increased when its traffic has a

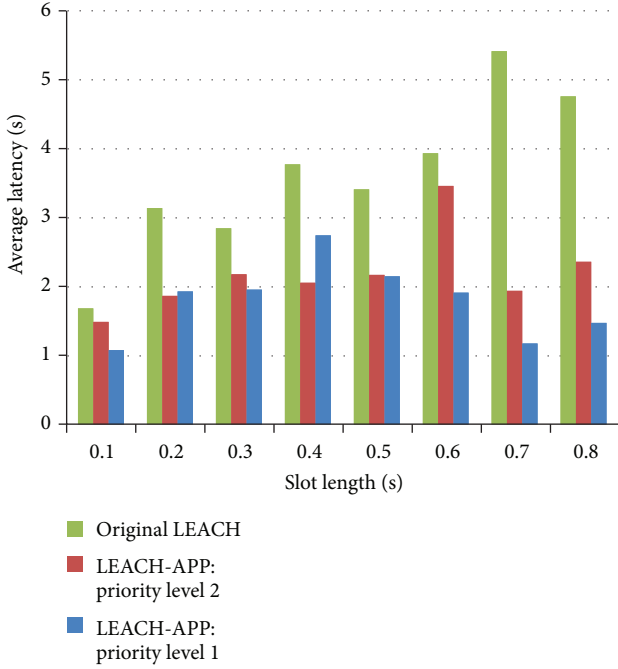


FIGURE 6: LEACH-APP vs. original LEACH: average latency of packets from source to sink.

higher priority level. This is true for all values of the slot length, although there are two cases where priority level 2 achieves a higher throughput than priority level 1. This is due to how the frames are formed and repeated throughout the round for these particular values of the slot length. In the most favorable case—slot length = 0.7 s—the throughput achieved with LEACH-APP supposes an improvement of roughly 250% over original LEACH.

It can also be clearly seen that the throughput has an inversely proportional relationship to the slot length. The reason for this is that, with a constant round length, increasing the slot length results in longer frames and consequently fewer frames per round. Depending on the number of nodes in the cluster, this can result in a lower total allotted time in the TDMA schedule. Also, nodes generate their packets at a constant rate and buffer them until they have an allotted slot to send them to the CH. When slots are farther away from each other, the possibility of an overflow in this buffer, and consequently packets getting discarded, is higher.

Figure 6 shows the average latency from source to sink of the packets in the above scenario. The effect of the proposed algorithm is analogous to the case of the throughput, with the packet latency effectively decreasing as the priority level is increased. In the most favorable case, also with a slot length of 0.7 s, LEACH-APP achieves a decrease in latency of almost 80% as compared to original LEACH.

In this case, there is no clear trend of latency increasing with the slot length as could be initially suspected. As we mentioned, having fewer frames per round can cause packet loss in some cases and a diminishing of throughput. However, since the latency is calculated only from the packets that arrive at the sink, this phenomenon has no impact in its value.

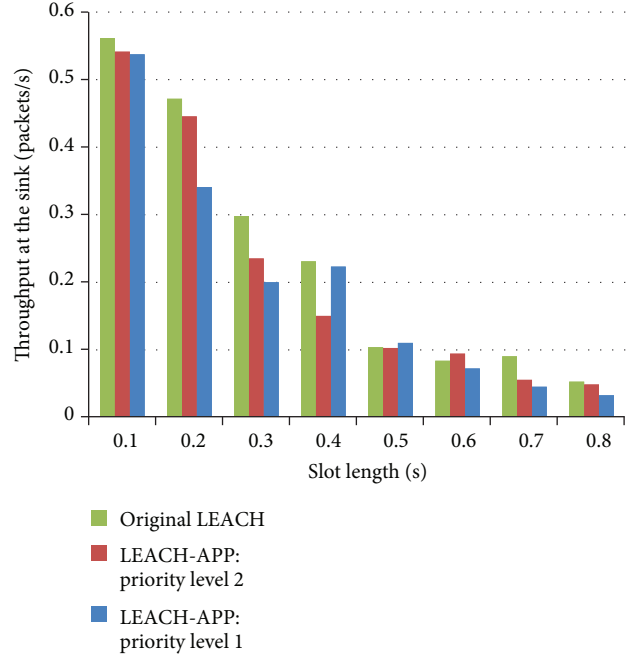


FIGURE 7: LEACH-APP vs. original LEACH: throughput of regular packets at the sink.

As mentioned in the case study of the previous section, improving the QoS of priority traffic has a negative impact on that of the regular one. Figure 7 shows the throughput of priority level 3 packets under the three configurations of the above scenario.

The graph shows that awarding extra slots to high priority traffic causes regular nodes to have fewer slots per round and subsequently a decrease in their throughput. The overall trend of throughput decreasing as the slot length increases happens for the same reasons explained above.

4.2.2. LEACH-APP: Impact of the Extra Slot Percentage Parameter. In this experiment, we evaluate the effect in the QoS of the main parameter in LEACH-APP: the percentage of extra slots assigned to each priority level. We consider again the baseline scenario under LEACH-APP, with one particular node (43) having priority level 1 and the remaining 99 having priority level 3. The slot length for this simulation is 0.2 s. Under these circumstances, we will evaluate the variations in throughput for different values of the extra slot percentage ESP1. As explained in the previous section, this parameter determines the percentage of extra slots allotted to PL1 traffic with respect to the number of cluster members. Figure 8 shows the results of this simulation.

In this particular scenario, the ESP1 parameter varies from 0 to 100% of the amount of nodes in the cluster. As this value is increased, the throughput rises too, but only until ESP1 reaches a value between 20 and 30%. Adding more extra slots from this point onwards has a negative effect in the throughput, yielding an even worse result than the case of adding none. The reason for this is again the fact that with a constant round length, adding extra slots results in longer

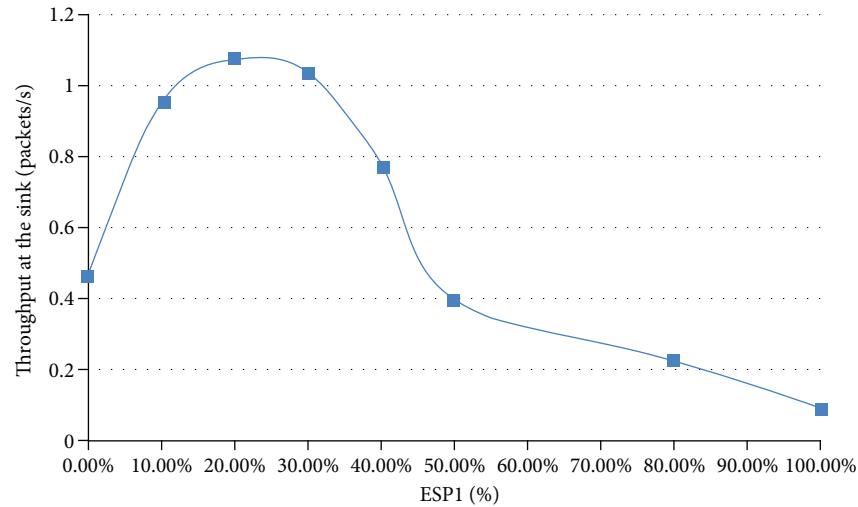


FIGURE 8: Impact of the extra slot percentage parameter.

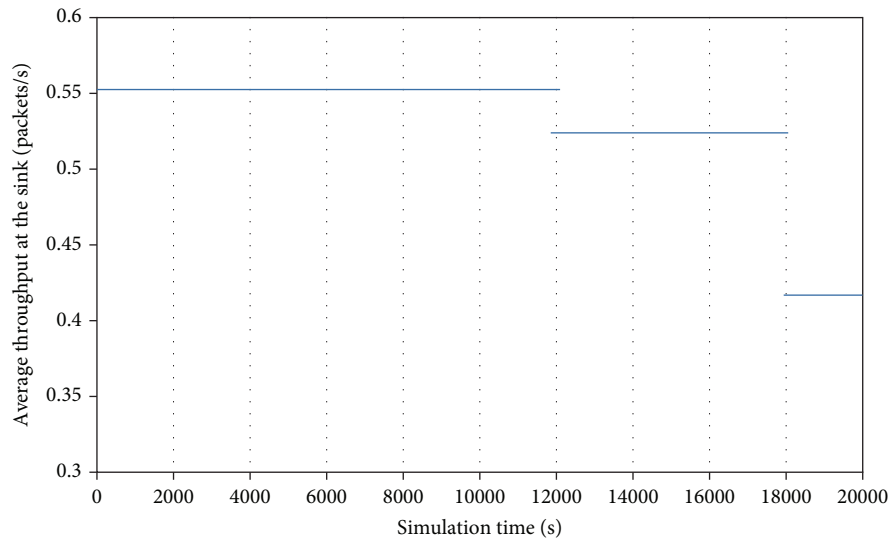


FIGURE 9: Changes in throughput in response to dynamic configuration.

frames and ultimately fewer frames per round and thus a lower slot repetition rate.

This experiment proves that the relationship between the throughput and the percentage of extra slots is not linear and has a saturation point. Thus, there is an optimum value of ESP1 to be sought depending on the particular conditions of each scenario.

4.2.3. LEACH-APP: Dynamic Configuration. This experiment is aimed at evaluating the dynamic nature of our proposed protocol. In LEACH-APP, the application can modify the priority level of any traffic stream at any moment, with changes taking effect in the following round. All the parameters from the basic scenario are used again, except for the simulation time which is increased to 20000 s.

Under this conditions, node 43 has a starting priority level of 1 until $t = 12000$ s, when the application lowers it to priority level 2. Then, in $t = 18000$ s, the application modifies

its level again from 2 to 3. Figure 9 shows the average throughput at the sink of packets from this node in these three periods.

This experiment could represent a realistic scenario in which the traffic of a particular node is very important to the application and consequently has a high priority from the beginning of the network's operation. However, after a certain time, node batteries could start depleting and energy efficiency becomes the top concern. In this situation, the application may desire to extend the network's lifetime, at the expense of reducing the resources allocated to this particular traffic, thus lowering its QoS.

5. Conclusions

In this paper, we have presented LEACH-APP, a new application-aware clustering routing protocol for WSNs aimed at providing better QoS. It is based on the original

LEACH protocol, arguably the most well-known in the field. Since different WSNs have different purposes and applications, we state the need for a scheme that not only is aware of these characteristics but also exploits them to achieve a better QoS.

Our protocol allows the user to classify the network's traffic into different priority levels according to their importance for the application. Then, in the set-up phase of LEACH, it assigns extra slots in the TDMA schedule to higher priority nodes. This way, the network provides more resources, in terms of allotted time, to higher priority traffic. Also, this can be modified dynamically, for instance attending to changes in the network conditions. Our scheme is integrated into the original LEACH operation seamlessly, with an almost negligible overhead.

In this paper, we present a thorough evaluation of the impact of our proposed contribution on two well-known QoS metrics: throughput and latency. We also compare the performance of our protocol with that of the original LEACH protocol. The simulations presented show that our protocol can provide a significant improvement in throughput, reaching increases of roughly 250% over original LEACH in certain cases. It also decreases latency by almost 80% in comparison to original LEACH for certain configurations of the simulated scenario. It is important to note that our protocol focuses on optimizing the QoS of selected traffic streams, not on fulfilling QoS constraints or providing deadline guarantees.

Although our protocol has a slightly negative impact in some QoS metrics of nonpriority traffic, this is done at the expense of giving the user freedom on how to allocate the network's resources. For a lot of WSN applications, it can be argued that this scheme provides a better overall QoS management than the usual case of letting the network run unattended. This is particularly true for the last part of the network lifetime, where nodes start depleting their battery and there is an overall degradation of performance.

Data Availability

The simulation data and set-ups used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This research was partially funded by the Spanish Ministry of Economy and Competitiveness, under RETOS COLABORACION program (reference grant: All-in-One: RTC-2016-5479-4) and CIEN program (ROBIM Project), and the Spanish Ministry of Industry, Energy, and Tourism through the Strategic Action on Economy and Digital Society (AEESD) under SENSORIZA: TSI-100505-2016-10 project. The authors also want to thank A. Pires and C. Silva from the Federal University of Para, Brazil, for their implementation of the original LEACH protocol in Castalia.

References

- [1] Cisco Visual Networking Index, *Forecast and Trends, 2017-2022 White Paper*, 2019, April 2019, <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>.
- [2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 10–19, Maui, HI, USA, 7 January 2000.
- [3] S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, vol. 5, pp. 4298–4328, 2017.
- [4] A. Rozas, J. Blesa, E. Romero, and A. Araujo, "Controlling the degradation of wireless sensor networks," in *Proceedings of 2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1217–1223, Dubrovnik, Croatia, 24–28 August 2015.
- [5] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [6] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [7] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [8] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [9] S. K. Singh, M. P. Singh, and D. K. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks," *International Journal of Advanced Networking and Application*, vol. 2, no. 2, pp. 570–580, 2010.
- [10] A. O. Fapojuwo and A. Cano-Tinoco, "Energy consumption and message delay analysis of QoS enhanced base station controlled dynamic clustering protocol for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 5366–5374, 2009.
- [11] B. Nazir and H. Hasbullah, "Energy efficient and QoS aware routing protocol for clustered wireless sensor network," *Computers and Electrical Engineering*, vol. 39, no. 8, pp. 2425–2441, 2013.
- [12] J. R. Diaz, J. Lloret, J. M. Jimenez, and J. J. P. C. Rodrigues, "A QoS-based wireless multimedia sensor cluster protocol," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, Article ID 480372, 2014.
- [13] S. Yahiaoui, M. Omar, A. Bouabdallah, E. Natalizio, and Y. Challal, "An energy efficient and QoS aware routing protocol for wireless sensor and actuator networks," *AEU International Journal of Electronics and Communications*, vol. 83, pp. 193–203, 2018.
- [14] V. K. Subhashree, C. Tharini, M. Swarna Lakshmi, L. E. A. C. H. Modified, and A. QoS-Aware Clustering, "Algorithm for wireless sensor networks," in *Proceedings of 2014 International Conference on Communication and Network Technologies*, pp. 119–123, Sivakasi, India, 18–19 December 2014.
- [15] E. Alnawafa and I. Marghescu, "New energy efficient multi-hop routing techniques for wireless sensor networks: static and dynamic techniques," *Sensors*, vol. 18, no. 6, pp. 1863–1883, 2018.

- [16] D. Pediaditakis, Y. Tselishchev, and A. Boulis, "Performance and scalability evaluation of the Castalia wireless sensor network simulator," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (ICST)*, pp. 1–6, Malaga, Spain, 16–18 March 2010.
- [17] A. Pires, C. Silva, E. Cerqueira, D. Monteiro, R. C. H. E. A. T. S. Viegas, and A. Cluster-Head Election, "Algorithm for WSN using a Takagi-Sugeno fuzzy system," in *Proceedings of 2011 IEEE Latin-American Conference on Communications (LATINCOM 2011)*, pp. 1–6, Belem, Brazil, 24–26 October 2011.
- [18] E. Casilari, J. M. Cano-García, and G. Campos-Garrido, "Modeling of current consumption in 802.15.4/ZigBee sensor motes," *Sensors*, vol. 10, no. 6, pp. 5443–5468, 2010.
- [19] A. Rastegarnia and V. Solouk, "Performance evaluation of Castalia wireless sensor network simulator," in *Proceedings of the 34th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 111–115, Budapest, Hungary, 18–20 August 2011.

