

Research Article

Self-Learning-Based Data Aggregation Scheduling Policy in Wireless Sensor Networks

Yao Lu ¹, Taihua Zhang,¹ Erbao He,¹ and Ioan-Sorin Comşa²

¹Technology Engineering Centre of Manufacturing Service and Knowledge Engineering, Guizhou Normal University, Guiyang, China

²Department of Computer Science, Brunel University London, Uxbridge, UK

Correspondence should be addressed to Yao Lu; luyaodada@163.com

Received 1 December 2017; Accepted 3 April 2018; Published 13 May 2018

Academic Editor: Biswajeet Pradhan

Copyright © 2018 Yao Lu et al. 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.

The problems of reducing the transmission delay and maximizing the sensor lifetime are always hot research topics in the domain of wireless sensor networks (WSNs). By excluding the influence of routing protocol on the transmission direction of data packets, the MAC protocol which controls the time point of transmission and reception is also an important factor on the communication performance. Many existing works attempt to address these problems by using time slot scheduling policy. However, most of them exploit the global network knowledge to construct a stationary scheduling, which violates the dynamic and scalable nature of WSNs. In order to realize the distributed computation and self-learning, we propose to integrate the Q-learning into the exploring process of an adaptive slot scheduling with high efficiency. Due to the convergence nature, the scheduling quickly approaches an approximate optimal sequence along with the execution of frames. By conducting the corresponding simulations, the feasibility and the high efficiency of the proposed method can be validated.

1. Introduction

As one of the most fundamental capability for WSNs, data collection aims to deliver the data generated from the monitoring area or object to the users who have interests on them [1]. The operations of wireless communication normally consume the most part of energy due to the electronic characteristics of sensor devices [2]. In this case, direct transmissions of original sensing data which contains redundant and noise information lead to high energy consumption and short nodal lifetime. To overcome this problem, data aggregation has been developed to merge data packets during the phase of data transmission, so only a small amount of meaningful information is conserved and delivered in the network. Consequently, the burden of involved nodes can be eased [3, 4]. The core concept of data aggregation demands that multiple packets have to reach the same intermediate node in a short period of time before performing aggregation function. It means that a special method is required to control the time point of communication. In the meantime, this method should ensure wireless communication with

less conflicts and collisions, which severely impacts network performance [5].

By being inspired from time division multiple access (TDMA), two mainstream methods are used to achieve the mentioned goals in WSNs [6]. The first category exploits the global knowledge of the entire network which may be variable in WSNs, such as topology information and electronic status of sensor device. After collecting these information, a stationary scheduling can be constructed in a centralized manner. However, this scheduling is fragile in the dynamic environment of WSNs [7]. For another category, the designers change the direction of implementation, the problem of finding the efficient scheduling is realized by the collaboration among multiple nodes, and the distributed computation only demands the local network information. Nevertheless, the existing methods of this category confront the problem of lacking an efficient guidance to approach the optimal solution [8]. For the purpose of concurrently inheriting the positive aspects of both categories, this paper provides a new scheduling policy, and the main contributions can be listed as follows:

- (i) We model the problem of exploring the optimal time slot scheduling in WSNs. Besides appending the energy consumption on the basis of other ordinary optimization objectives in wireless communication, some specific constraints are equally considered in order to apply data aggregation in the phase of transmission.
- (ii) By exploiting the computation capability of sensor device, the global scheduling optimization problem can be transformed into distributed collaborative tasks on sensors, and this feature makes the proposed approach easily accommodated for dynamic network environment conditions of WSNs.
- (iii) Q-learning is used into the discovery process of the optimal scheduling by utilizing the local knowledge from the neighbour environment. The selection process of the time slot is formulated as a limited state machine [9].

The rest of the paper is organized as follows: Section 2 analyzes the characteristics of the existing approaches. Section 3 states the problem which we are concerned and clearly defines the problem of locating the optimal scheduling. In Section 4, the main task is to describe the principle and procedure of a novel scheduling policy with self-learning feature. Section 5 briefly introduces the simulation platform and explains the simulation result in order to prove the high performance of the proposed policy. Finally, the conclusion and the future work are put in Section 6.

2. Related Work

Without considering applying data aggregation to save energy, the previous slot scheduling policies focus only on decreasing transmission delay and collision-free wireless communications [10, 11]. To match the dynamic feature of WSNs, a distributed self-learning scheduling approach (DSS) is applied in [12]. The principle of Q-learning is implemented with the exploration process of a near-optimal time slot scheduling. The distributed Q-learning makes its implementation on sensor nodes very easy. Nevertheless, the performance of this approach on energy consumption could be unsatisfactory due to the lack of data aggregation at the network and MAC levels. This may be unacceptable for some applications which pay more attention on energy consumption and nodal lifetime [13].

To achieve the medium access control for single-hop wireless sensor network, frame-based ALOHA protocol is firstly proposed in [14]. By importing the concept of Q-learning, the communication slots of each frame can be selected in an intelligent manner. The slot selection naturally migrates from random access to perfect scheduling in steady state conditions. The convergence property of this method is validated by a Markov model of the learning process. Nevertheless, the assumption of single-hop communication severely restricts the application range of this method. Data aggregation function is not considered during data transmission as

well, then this method only attempts to save energy by decreasing the communication collision.

A centralized aggregation scheduling policy called Peony-tree-based data aggregation (PDA) is proposed in [15]. In the network, nodes are subdivided into multiple levels based on the hop count information, where the base station is located at the first level, and the leaf nodes stay at the bottom level. By abstracting the wireless sensor network as a graph, a maximal independent set can be created in order to help constructing the data aggregation tree. Besides the aggregation efficiency, the most important mission is to reduce the aggregation latency. In this case, the leaf nodes have to be first scheduled, and then the dominator and connector in maximal independent set are scheduled level by level. Although the aggregation freshness and data accuracy can be ensured in PDA, the construction of maximal independent set demands extra overhead of computation and communication.

In a large number of WSN application cases, the energy consumption and transmission delay are normally two conflicting optimization objectives, and an efficient scheduling is supposed to achieve a good trade-off between them. Nearly constant approximation (NCA) for data aggregation scheduling in WSNs is developed to solve this multiple-objective scheduling problem [16]. It assumes the network is static and collects the necessary global environment information. After obtaining these information, a powerful node executes the search algorithm to find out a high efficient solution. However, the assumption cannot be always met in WSNs due to the change of the network state.

Even though the problem to be solved is the same as NCA, distributed delay-efficient data aggregation scheduling (DAS) provides a complete different approach [17]. The computation capability of each sensor device is exploited, and they coherently cooperate to find a near-optimal slot scheduling. The construction process of scheduling is independent with the phase of data transmission, and a sensor node forcedly maintains a limited state machine with at least five states to realize its construction task. In addition, the usage of tree level as the main index to reduce the aggregation delay could be inaccurate in some application cases, and it may make the final solution far away from the theoretical optimal solution.

3. System Model and Problem Statement

3.1. System Model. Let us consider a WSN as a directed incomplete graph $G(N \rightarrow \overline{E})$, where N is a finite set of sensor nodes, which are uniformly or randomly distributed in the areas of monitoring regions and responsible to generate measuring data. If the sensor nodes $n_i \in N$ and $n_j \in N$ have a direct communication, the directed links $\overrightarrow{E}_{i,j}$ and $\overrightarrow{E}_{j,i}$ both existed due to the symmetric communication. In order to realize a duty cycle WSN, we assume all nodes in the network have appropriate clock synchronization. The lifetime of each node consists of multiple frames with the same length, and each frame is composed by F time slots. Based on the rule of time slot scheduling, a node can switch its own state between

ACTIVE and SLEEP. If a node keeps ACTIVE in a certain time slot, then one transmission or reception can be performed. Otherwise, a node temporarily switches off its power supply for primary electronic units in the SLEEP state, and this behaviour contributes to effectively save more energy.

According to the feature of wireless communications, if there is a transmission performed on the link $\overrightarrow{E_{i,j}}$, then the length of this link $r = |n_i, n_j|$ is actually the radius of the transmission disc of node n_i and n_j . The other nodes or neighbours located in this disc are interfered by this communication. If any other transmission is occurring in this area, the communication collisions will subsequently happen. Besides, one node can only choose to transmit or receive data packet in one ACTIVE slot. Once a data packet is successfully transmitted to a receiver, the corresponding transmitter is supposed to get an ACK message from this receiver in the same slot. However, if the transmitter does not obtain ACK, then this packet will be combined with the packet in the next frame and retransmitted. Since the WSN routing protocols do not belong to the research scope of this paper, we assume that the routing structure of data aggregation has been already constructed. The nodal relationship is determined for each node, which means that a node has the knowledge about its upstream and downstream nodes. To simplify the model definition, the complete aggregation function which could combine multiple packets into a single packet is applied in this paper. Therefore, only one transmission can be performed in one frame.

3.2. Problem Statement. From the viewpoint of the entire network, each node in the routing structure is supposed to perform transmission once in one frame, let us assume F_a denoting the actual number of slots used for data transmission in one frame, and then, a sequence of the transmitter sets $AS = \{TS_{t_1}, TS_{t_2}, \dots, TS_{t_{F_a}}\}$ can represent a data aggregation scheduling, where TS_{t_i} denotes the nodes that need to transmit data in the slot t_i . F_a also means that the base station will receive all data at the F_a th slot. For the purpose of decreasing the aggregation delay, we have to let the user acquire all the demand data as soon as possible. Reducing F_a becomes one of the optimization objectives for aggregation scheduling policy. The number of communication collisions C_n should be minimized as well, and the ideal condition is collision-free where this number is equal to zero. In the meantime, a sensor device is sensitive to its own energy consumption. Therefore, an efficient scheduling policy should make the average value of energy consumption E_c^{avg} as small as possible, and its value normally depends on the number of SLEEP slots and the number of communication collisions:

$$\begin{aligned} & \arg \min f(F_a(AS), C_n(AS), E_c^{avg}(AS)) \\ \text{Constraints :} & \\ & TS_i \cap TS_j = \emptyset, \quad \forall i \neq j, \\ & ST(n_i) > \max\{SR(n_i)\}, n_i \text{ is not leaf node.} \end{aligned} \quad (1)$$

In addition to the optimization objectives, there are still some constraints for constructing an efficient slotted scheduling of data aggregation. Based on the property of single transmission in one frame at each node, any TS_{t_i} should be disjointed. Besides this, for a node n_i , let us suppose that the active slots for data aggregation and transmission is $ST(n_i)$, and the set of active slots for reception is $SR(n_i)$. Depending on the principle of data aggregation, transmissions have to be activated after the last reception slot. In the case when an aggregation scheduling is defined as AS, the theoretical optimal aggregation scheduling can be expressed as shown in 1. Parameter f denotes the evaluation function of aggregation scheduling, and the first constraint corresponds to the limitation of transmission number on each node, and the second constraint matches the requirement of data aggregation. In order to clearly explain the mentioned concepts of aggregation scheduling, we use an example in Figure 1. A routing structure rooted from the sink terminal T is constructed before exploring an efficient scheduling. $F = 6$ means that one frame is composed of 6 slots, and $F_a = 4$ means that data can be transported to the sink node only by using 4 slots. $TS_{t_1} = \{n_a, n_d\}$ because the terminals $\overrightarrow{E_{a,e}}$ and $\overrightarrow{E_{d,f}}$ have no conflicts during the simultaneous transmission. The last reception slot of n_f is $\max\{SR(n_f)\} = t_2$ which is smaller than its transmission slot $ST(n_f) = t_3$.

3.3. Distributed Solution Based on Q-Learning. The problem of discovering the theoretical optimal aggregation scheduling has been proven as NP-hard [18]. Plenty of existing methods are utilizing the global knowledge of the entire network to find an approximate solution. But, in order to collect the necessary knowledge, it will violate the dynamic nature of WSNs, and it will make these methods inapplicable in many cases. Once the value of any main parameters in network environment makes some changes, new information has to be gathered again and this process cost too many resources. Instead of that, we introduce a new distributed aggregation policy which benefits from the idea of reinforcement learning. The exploration burden has been assigned to sensor nodes. Decision of whether a time slot is active or not is described as a Markov decision process (MDP). By using the local knowledge such as feedback from neighbours, an approximate optimal scheduling of data aggregation can be found.

4. Dynamic Aggregation Scheduling Policy

4.1. Preliminaries on Reinforcement Learning. The goal of reinforcement learning is to guide the selection of ACTIVE slots for either reception or transmission without a priori knowledge. To follow this idea, a scheduling process of slots should be expressed by using the MDP model. According to the definition of MDP, there is a tuple with four elements (S, A, P , and R), where S is the state space, and A contains all potential actions at each state. $P: S \times A \times P \rightarrow [0, 1]$ denotes the transition probability of states by taking a certain action. R is the reward function which depends on $S \times A \rightarrow R$. In the original Markov model, the final goal for an agent is to

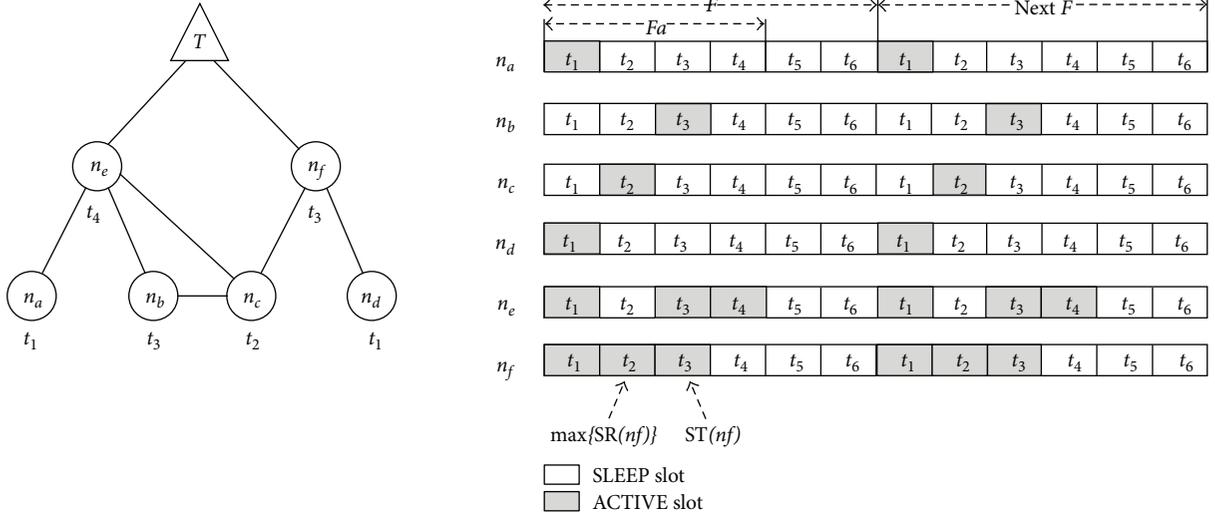


FIGURE 1: Example of aggregation scheduling.

maximize the expected discounted reward or state-value, and it can be expressed as

$$v^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} p_{ss'}(\pi(s)) v^\pi(s'), \quad (2)$$

where $v^\pi(s)$ represents the value of objective function, π is the adopted action policy, $R(s, \pi(s)) = E\{r(s, \pi(s))\}$ denotes the expected value of reward, γ is the discount factor and s to s' . In our case, it is difficult to find out $p_{ss'}(\pi(s))$ as the transition probability from state $p_{ss'}(\pi(s))$ and $R(s, \pi(s))$. Therefore, Q-learning becomes a feasible algorithm for learning the delayed reinforcement information to locate the optimal solution. Correspondingly, the objective function and the action policy are transformed into a two-dimensional table which uses state-action pairs as the index and Q-value in each element. If we assume that the learning rate is α , so the update rule of Q-value can be expressed as follows:

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha \left(r_k(s, a) + \gamma \max_{a' \in A(s')} (Q_k(s', a') - Q_k(s, a)) \right), \quad (3)$$

where $Q_{k+1}(s, a)$ denotes the Q-value at the state s with the action a , and $r_k(s, a)$ represents the current reward or the immediate reward. In this way, the action policy π can be denoted by the following equation:

$$\pi(s) = \arg \max_{a \in A(s)} (Q(s, a)). \quad (4)$$

Before smoothly applying Q-learning in WSN data aggregation scheduling process, it is necessary to match the system model of time slot scheduling with Q-learning approach and specify the corresponding components of MDP in the process of slot selection.

4.1.1. State and Action. Each single sensor device is treated as an agent. The selection process of one ACTIVE slot for receive or transmit is considered as an MDP. The state and action are the selected ACTIVE slots in the previous frame and the current frame, respectively.

The knowledge of upstream and downstream nodes are obtained before automatically scheduling slots. The previous state of selection has very limited influence on the action of selecting the current slot; therefore, only one row is conserved in Q table without splitting by different previous state in order to relief the computation burden. Let us assume the number of upstream nodes is N_R , the same number of reception Q tables are generated. The reception slots from different upstream node are selected from their corresponding tables in order, where $\text{Tab}_{Q_R}^i$ denotes the reception table for i th upstream node. The same slot cannot be exploited for different upstream nodes. To comply with this rule, the slots which are used by the previous upstream node should be excluded from the candidate set of slots for subsequent upstream nodes. To maximize the effect of data aggregation, the transmission of aggregated output always appears after the receptions of all expected packets. In this case, the candidate transmission slot is supposed to be located behind all reception slots which are selected in this frame, and the transmission Q table is represented by Tab_{Q_T} . For each node, ST_c is the current transmission slot, and SR_c is the current set of selected reception slots in one frame, where $\text{SR}_c = \{t_x, \dots, t_y\}$. SR_h is the combination of multiple historical SR_c in recent h frames, where $\text{SR}_h = \{\text{SR}_c^1, \dots, \text{SR}_c^h\}$.

An example of a Q table is depicted in Figure 2. For current node n_k , there are two upstream nodes n_i and n_j , their corresponding reception tables are $\text{Tab}_{Q_R}^i$ and $\text{Tab}_{Q_R}^j$, respectively. The downstream node and transmission table are n_l and Tab_{Q_T} , respectively. When the process of active slot selection proceeds, these slots are choose sequentially from these tables. In the first step, t_3 is the reception slot for the first upstream node, it has to be disable as the candidate slot

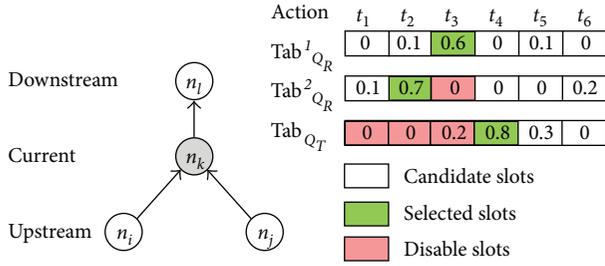


FIGURE 2: Example of a Q table.

for other selection, and then t_2 is selected as the second reception slot. After all reception slots are decided, the candidate set of transmission slot begin from the last reception slot, then t_4 becomes the transmission slot. In this case, $\text{SR}_c = \{t_2, t_3\}$ and $\text{ST}_c = t_4$.

4.1.2. Reward Function. According to the difference of reception and transmission, the implementations of their reward function are obviously different.

(1) Reward Function of the Reception Q Table. By observing the optimization objectives of slot scheduling in 1, the corresponding operations will be embodied in the process of Q-value updating. For the purpose of minimizing the delay, a node should be inclined to choose the feasible slots with the smaller sequence number or placed in the anterior parts of one frame. Therefore, the position of the last working slot is advanced in the entire frame, and a much lower delay can be achieved. In the update equation, a delay factor will be utilized to influence the Q-value. In order to avoid the communication collision, reward and punishment are used to deal with different reception situation. If a packet is successfully received in an ACTIVE slot, then the Q-value will be increased. In another case, no packet is received in an appointed reception slot, and some punishments should be acted on its Q-value. In the last case, the energy consumption is represented by the energy factor, which is basically the combination of two impact parameters, such as the maximum retransmission number, the product of distance, and packet size. Retransmission is the waste of energy, because the transmitter has to consume extra power in order to transport the same data again. So, retransmission number should be minimized. The product of distance and packet size generates the preference of receiving packets with larger size from farther sources. According to the theory of energy consumption in [19], this selection preference can effectively save more energy.

For the purpose of formulating the reward function, let t_i represent the i th slot in one frame, n_a be the maximum number of retransmission attempts, h_d be the hop count towards the destination at d th transmission for a packet, and k_d be the packet size of the d th transmission. Since the unit of packet size may severely weaken the impact of other factors, then it has to be normalized as follows:

$$\bar{k}_d = \frac{k_d - k_{\min}}{k_{\max} - k_{\min}}, \quad (5)$$

where \bar{k}_d is the normalized packet size. If we assume that a packet is received at t_i as $pr(t_i) = 1$ and the contrary circumstances as $pr(t_i) = 0$, then the reward function can be rewritten as follows:

$$r_R(t_i) = \begin{cases} \frac{\delta(F-i)}{F} + (n_a + 1) \sum_{d=2}^D (h_d - h_{d-1}) \bar{k}_d, & pr(t_i) = 1, \\ -\theta \frac{i}{F}, & pr(t_i) = 0, \end{cases} \quad (6)$$

where δ is the impact factor of the slot position, D is the sequence number of current transmission, and finally, θ is the punishment factor that ensures the effect of punishment. The usage of different factors for reward and punishment functions can be attributed to two reasons. Firstly, some parameters contained in the energy factor are unobtainable when there is no received packet. Secondly, the slot placed in the front position of the entire frame is always preferred by the selection of subsequent frames. In this case, its increment of reward is larger and decrement of punishment is smaller than the slots located behind it.

To clearly explain the updating process, an example is depicted in Figure 3. The middle table displays the values before updating, where slot t_3 is selected by the previous upstream node, so the column of action t_3 should be disabled. Slot t_2 has the highest Q-value in the table, then it is selected as the reception slot for the current upstream node. If a packet is surely received at this slot t_2 , then a reward will be given to it, as it was the case in the right table. Otherwise, the Q-value of slot t_2 is reduced due to its punishment.

(2) Reward Function of the Transmission Q Table. To keep the consistency with the optimization objective in 1, the slot with smaller sequence number is still preferred to be used. Besides this aspect, the product of retransmission number and aggregation efficiency will embody the energy consumption. Scheduling policy tends to pick the slot which successfully transports a packet with high retransmission number, because this behavior effectively relief the network burden. The aggregation efficiency is the ratio of the receiving data size and the aggregating data size, and the higher efficiency should correspond to the higher Q-value, which is encouraged to be selected.

The definition of abovementioned parameters is inherited, and then, let us consider the size of reception and aggregation to be represented by k_{rec} and k_{agg} , respectively. Let us further suppose that the reception status of ACK at t_i is represented by $pra(t_i)$. The assignment $pra(t_i) = 1$ means that ACK is received; otherwise, $pra(t_i) = 0$. Consequently, the updating rule of transmission slot can be expressed as follows:

$$r_T(t_i) = \begin{cases} \frac{\delta(F-i)}{F} + (n_a + 1) \left(1 - \frac{k_{\text{rec}}}{k_{\text{agg}}}\right), & pra(t_i) = 1, \\ -\theta \frac{i}{F} - (n_a + 1) \left(1 - \frac{k_{\text{rec}}}{k_{\text{agg}}}\right), & pra(t_i) = 0, \end{cases} \quad (7)$$

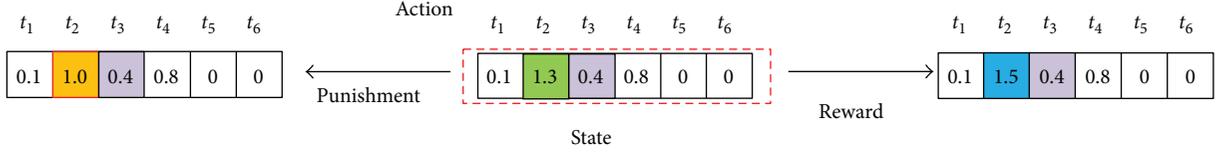


FIGURE 3: Example of Q-value updating for reception slot.

where $k_{\text{agg}}/k_{\text{rec}}$ actually represents the ratio of aggregation. If this value is higher, the aggregation effect is more obvious.

4.1.3. Action Policy. In order to keep the random searching capability of Q-learning, we adopt a variant action selection policy $\Delta\epsilon$ -greedy instead of the standard ϵ -greedy, where a random slot can be selected with the narrowing probability and the best slot with the maximum Q-value can be used otherwise. The searching range should be larger at the beginning of exploration or learning, and the range should be narrowed with the convergence of exploration. Based on this principle, if the current sequence number of frame is F_{sn} and ρ is the shrinking factor, then $\Delta\epsilon$ can be expressed as follows:

$$\Delta\epsilon = \frac{e^{-F_{\text{sn}}}}{\rho(1 + e^{-F_{\text{sn}}})}. \quad (8)$$

In this way, alongside with the increase of the frame number, the probability to use random selection becomes very small. This behaviour contributes to a quick convergence of the slot selection.

4.2. Implementation of Time Slot Scheduling Based on Reinforcement Learning. Through observing the working process of time slot scheduling, the procedure of automatically coordinating a great number of sensor nodes to achieve efficient communication becomes distinct. Generally speaking, there are two primary tasks for the scheduling policy. First, the active slots which are supposed to receive and transmit packets should be decided before the execution of a new frame. Second, each slot of one frame starts to work sequentially, once the state of a slot is ACTIVE, the corresponding actions are subsequently performed.

4.2.1. Node Behaviour on Time Slot Scheduling. In Algorithm 1, from line 2 to 7, the current node is supposed to receive N_R packets from the same number of upstream nodes, and the slot with the highest value for each reception is selected from the independent Q-value tables. The result is recorded into the current set of selected reception slots. Once the last reception slot is decided, the transmission slot can be selected from the last reception slot to the last slot in the frame. In the next step, the selection results with historical information are compared in lines 11–15. A condition of judgment is utilized to distinguish whether the selection result is stable or not, and the principle of this condition will be explained in the following paragraph.

The execution of slots starts with line 16. From lines 1724, the problem of handling the transmission slot is described. For each slot, if the state is ACTIVE, and the current slot is allocated to transmit, the aggregated packets will be transmitted. Afterwards, the system expects to receive

ACK. The value of this slot t_j in the Q table Tab_{Q_T} will update depending on whether ACK is received or not. In lines 2527, if the packet from the i th upstream node is received, then the value of this slot in the corresponding Q table $\text{Tab}_{Q_R}^i$ is reinforced with rewards. However, if the current slot is assigned to receive the packet from a specific upstream node, but there is no packet arriving, then its Q-value will be decreased by using punishment. Besides, several subsequent slots are activated in order to reduce the opportunity of unsuccessful communication. This process is depicted in lines 28–30.

4.2.2. Stable Condition of Selection Result. The reason to distinguish the state of selection result originates from the goal of balancing the energy consumption and the successful transmission ratio. Depending on the judgment result, a node can adjust its own strategy to transit more slots from SLEEP to ACTIVE state. In case of STABLE state, the upstream nodes have high probability to choose the same slots for transmitting data, and other slots can be switched off to further save energy. Otherwise, too many SLEEP slots may cause low successful transmission ratio, but it is uncertain when the incoming packets will arrive at current node. Keeping all slots as ACTIVE can definitely increase the probability for the successful data transmission. Based on the definition of SR_h , whether the selection of slots is stable can be confirmed by using the similarity index, and this metric depends on the comparison result of each two successive selection sets:

$$\bar{J}(\text{SR}_h) = \sum_{h=1}^{i=1} w_i \frac{|\text{SR}_c^i \cap \text{SR}_c^{i+1}|}{|\text{SR}_c^i \cup \text{SR}_c^{i+1}|}, \quad (9)$$

where $\bar{J}(\text{SR}_h)$ denotes the similarity index, and is the weight of the similarity of two sets. Once the similarity index is larger than a predefined threshold, the state of selection can be viewed as stable. Since the tendency of stableness makes the comparison result of the latest two successive selection sets more important than the historical result, then more weight should be allocated to the latest comparison results. The weight can be defined as

$$w_i = \frac{2i}{h(h-1)}, \quad (10)$$

which can ensure the latest results having higher weight, and $\sum_{i=1}^{h-1} w_i = 1$. This switchover between two states can markedly avoid the phase with low communication efficiency.

4.3. Adaptation for Dynamic Network Environment. Dynamic network environment is one of the most important characteristics of WSNs. Since sensors are normally deployed in

```

Data:
 $N_R, SR_c, Tab_{Q_R}^i, Tab_{Q_T}, SR_h$ 
1  while a new frame starts to execute do
2    for  $i = 1; i \leq N_R; i++$  do
3      for  $k = 1; k \leq |SR_c|; k++$  do
4         $Tab_{Q_R}^i$ .disable( $SR_c(k)$ );
5      end
6       $t^R = \Delta\epsilon - greedy(\text{argmax}_{t^R \in SR_c} \{Tab_{Q_R}^i(t^R)\})$ ;
7       $SR_c$ .add( $t^R$ );
8    end
9     $SR_h$ .add( $SR_c$ );
10    $ST_c = \Delta\epsilon - greedy(\text{argmax}_{t^T > SR_c.last()} Tab_{Q_T}(t^T))$ ;
11   if  $SR_h.comparison()$  is not stable then
12     Set the state of all slots as ACTIVE;
13   else
14     Set  $t_k$  as ACTIVE where  $\forall t_k \in SR_c$ , and set other slot as SLEEP
15   end
16   for  $j = 1; j \leq F; j++$  do
17     if the state of  $t_j$  is ACTIVE then
18       if  $ST_c == t_j$  then
19         send packet and wait for ACK;
20       if ACK is received then
21         Update  $Tab_{Q_T}(t_j)$  with rewards;
22       else
23         Update  $Tab_{Q_T}(t_j)$  with punishment;
24       end
25     else if receive from  $i^{th}$  upstream node then
26       Update  $Tab_{Q_R}^i(t_j)$  with rewards;
27     end
28     else if  $SR_c(i) == t_j$  and not receive any packet from  $i^{th}$  upstream node then
29       Update  $Tab_{Q_R}^i(t_j)$  with punishment;
30       Set  $t_{j+1}$  to  $t_{j+p}$  as ACTIVE;
31     end
32   end
31 end
32 end

```

ALGORITHM 1: Node behaviour on time slot scheduling.

complicated external environment and their resources are limited, any change on the sensors' state or the surrounding environment may cause the sensor to malfunction or lost communication. Consequently, network topology may be not always stationary, for example, a sensor node suddenly becomes disconnected, or a node participates into the network in order to cover more monitoring area. Node disappearance and appearance are very common situations in WSNs, and the proposed scheduling policy is able to automatically handle these situations without repeatedly gathering information from the global network.

In general cases, a node that disappears in the network will only affect the routing and scheduling relationship in local range. It is not necessary to make a reaction in the global network. The disappearance may be caused by many reasons, such as battery exhaustion and malfunction. We do not concern the problem of how to detect the disappearance, then we assume that a node is able to send a disappearing notification

message to neighbour nodes before a formal disappearance. For a neighbor node, once its i^{th} upstream nodes disappear, then N_R and SR_h are subsequently updated, $Tab_{Q_R}^i$ is removed as well. Q-value in remaining tables of reception and transmission is not affected in order to conserve the learning memory. However, $\Delta\epsilon$ of Tab_{Q_T} has to be set as the original value and gradually grow with the number of frames. The objective is to recover the random searching capability. Consequently, better slots are probably found when comparing with previous selections. If the unique downstream node disappears, an alternative downstream node will be picked by the routing protocol. Then, all values of Tab_{Q_T} are forcedly to be reset due to the previous slot which is not feasible for wireless communication anymore. This principle is inherited from the reaction process of appearing a new node in the network. When a new node intends to participate in the current network, it broadcasts an appearance notification to neighbor

nodes. The routing strategy of neighbour nodes decides the transmission relationship with this new node. After that, the scheduling policy starts to find out the suitable slots. No matter how this new node performs as upstream node or downstream node for the current node, the transmission slot is supposed to be determined again.

5. Simulation and Performance

OMNeT++ is the simulation platform used to evaluate the performance of the proposed scheduling policy. MiXiM is a common modelling framework for wireless communication based on OMNeT++ [20]. Through utilizing this layered model, different tasks of sensors can be implemented at different network layer. As one of the most common application in WSNs, periodical data collection event is implemented on the network layer. The guide of the transmission direction of data packet is the job of the routing protocol. The corresponding implementation is put on the network layer. Finally, the control of communication time point is completed by the scheduling policy on MAC layer. Other necessary components are provided by the MiXiM framework. The routing protocol based on data aggregation is not the concern of this paper, and a conventional method called energy-aware spanning tree [21] is used to construct the routing structure. In this case, the proposed DSQ (data aggregation scheduling policy based on Q-learning) gets the node relationship before allocating the suitable time slot for each node.

The performance of scheduling policy is highly related to the configuration of system parameters. In the simulation, even though the deployment of sensor nodes is a random distribution, the connectivity of network is ensured. The common application scenarios with different network size and different number of source nodes should be considered. Besides, the setting of parameters is recommended by conducting many a priori experiments, where $\alpha \in [0.1, 0.4]$, $\gamma \in [0.1, 0.2]$, $h = \{4, 6, 8\}$, $\delta \in [1.2, 1.6]$, $\theta \in [2.0, 10.0]$, and $k_{\min} = 100$ and $k_{\max} = 5000$.

For the purpose of embodying the advantage of the proposed scheduling method, three existing methods with different types are selected to do the contrast. DSS (distributed self-learning scheduling approach) is a slotted scheduling policy without considering data aggregation, which aims to realize the low delay and collision-free communications in WSNs. NCA (nearly constant approximation) imports the concept of data aggregation into time slot scheduling, but it requires the global knowledge of entire network and performs the centralized computation for the scheduling plan. DAS (distributed delay-efficient data aggregation scheduling) combines the technique of data aggregation and distributed implementation, but its efficiency is not always satisfied due to the lack of the global viewpoint of the network.

Based on the objectives of scheduling policy in 1, three main metrics are adopted to exhibit the performance of these comparative methods. Transmission delay can be observed by the average duration between the transmitting time point of a packet at the source node and the receiving time point of this packet at the sink node or destination node. The average number of communication collisions can be easily counted

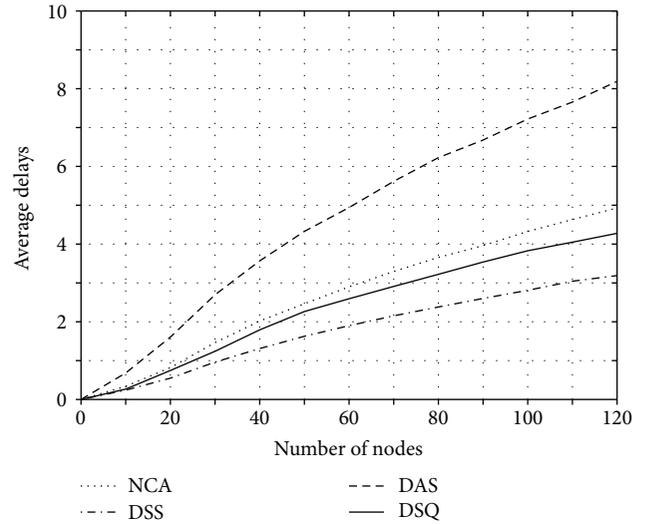


FIGURE 4: Average delay with different network size.

by using the network simulator, which can embody one important aspect of communication quality. Average residual energy of the involved transmission nodes is the direct index of nodal lifetime, and it can reveal the effect of data aggregation.

In Figure 4, four scheduling policies are compared. DAS gets the highest delay when comparing with other methods, and the gap becomes more apparent as the increase of network size. In the worst case, DAS takes 2.6 times of delay when comparing with DSS. It does not find an efficient optimization mean to make the scheduling approaching the theoretical optimal solution. Since DSS gives the first priority to the transmission delay, it has the best performance on this metric. DSQ treats delay as the main optimization objective and prefers the slots ahead in the scheduling, so it performs a little bit better than NCA, and it saves about 19% of delay in the best case. In the meantime, DSQ only requires half of DAS's delay to transport the same amount of data. The number of frames also has impacts on transmission delay for different methods as shown in Figure 5. DSQ has the learning feature, it costs a short period to be stable and locates an approximate best solution, and the average delay is reduced with 70% when the number of frames increases to 104. In this case, DSQ takes longer delay when comparing with NCA at the few number of frames. This is due to the fact that it still attempts to find a good scheduling. When the more frames are executed, the scheduling of DSQ becomes convergent, and it reaches a lower delay. The delay of other three policies has barely changed due to their advanced scheduling construction before data transmissions. Meanwhile, DSQ has the highest value of deviation due to the mutative nature at the beginning of execution.

The average residual energy of the involved transmission nodes is depicted in Figure 6, and its unit is set as percentage instead of joule. When the number of source nodes increases, the energy level of DAS dramatically decreases, because data aggregation is not considered. DSQ conserves about 1.3 times more energy than DAS when the largest number

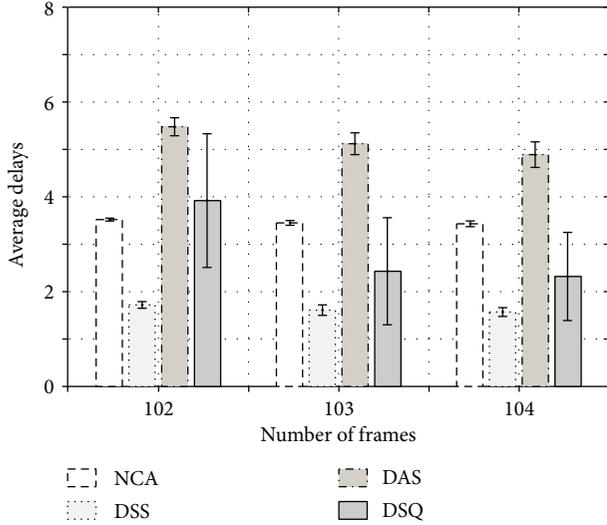


FIGURE 5: Average delay with different number of frames.

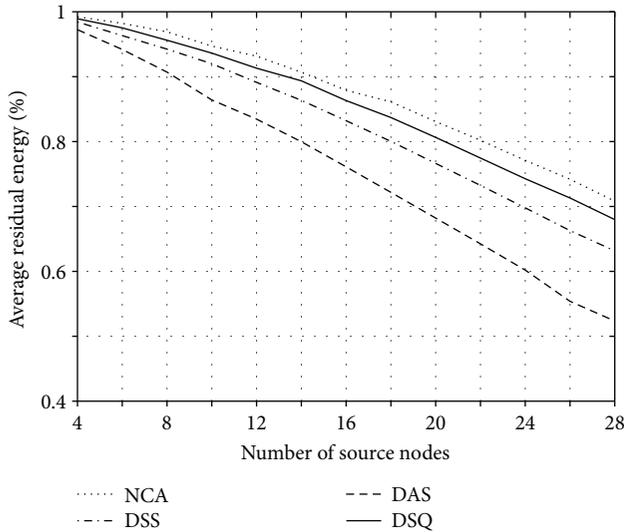


FIGURE 6: Average residual energy with different transmission task.

of source nodes is considered in transmissions. The feature of centralized data aggregation scheduling contributes to keep the highest level of energy by NCA. Along with the change of frame numbers, DAS quickly drops to a very low energy level due to that fact that it delivers too many packets as depicted in Figure 7. When the frames grow from 102 to 104, the energy level of DAS decreases to 31.6% of the original value. DSQ aims to reduce the energy consumption by using data aggregation, and it is able to conserve energy more 1.76 times than DAS and 1.21 times more than DSS. Thanks to the global searching capability for the optimal aggregation scheduling, NCA obtains the best performance.

Being different with other two aggregation scheduling policies, DSQ does not occupy an independent period to construct a stationary time slot scheduling. Instead of that,

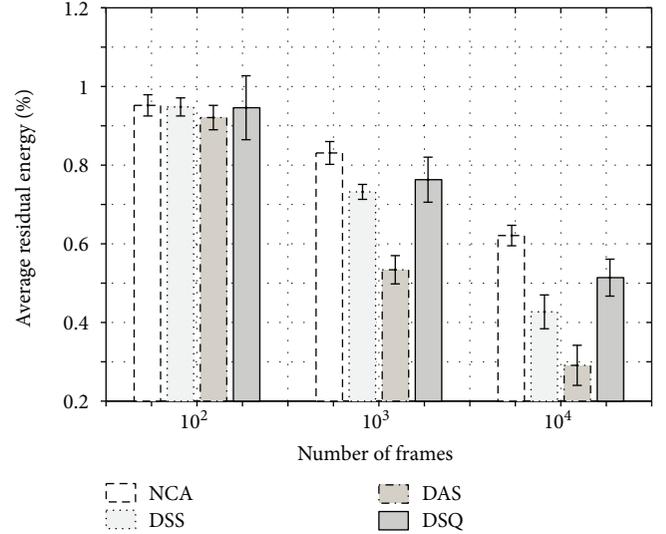


FIGURE 7: Average residual energy with different number of frames.

TABLE 1: Average number of communication collisions.

Frames	Number of source nodes			
	10	15	20	25
10^2	14.23	17.39	26.56	43.16
10^3	3.31	6.25	9.44	16.92
10^4	0.47	0.87	1.22	1.78

an approximate optimal scheduling is supposed to be discovered among the ordinary data transmission. Therefore, the sequence of scheduling is dynamic at the beginning of exploration, and then quickly converge to a stable sequence with high quality. The benefit of this feature is to save the time for independent construction and automatically adapt to the dynamic network environment. The average number of communication collisions for each frame is depicted in Table 1. Since DSQ has the convergent nature of making the scheduling sequence approach the optimal solution, communication collisions are remarkably reduced when the number of frames increases. The most obvious result can reach 3% of the original collisions with 100 frames.

The problem of how to clearly observe the convergence process of slot selection can be solved by using a special metric. The general principle of this metric SC is to find out the selection consistency among recent F_r frames, where $F_r = 10$. Let the selection result of the frame i at node k be S_i^k , then its function can be defined as follows:

$$SC = \frac{1}{(F_r - 1)N} \sum_{k=1}^N \sum_{i=F_r}^{F-1} \frac{|S_i^k \cap S_{i+1}^k|}{|S_i^k \cup S_{i+1}^k|}. \quad (11)$$

The corresponding result of the selection consistency can be found in Figure 8. Generally speaking, the consistency will gradually approach to the maximum value of 1 along with the execution of more frames. The variation trend of this metric embodies the convergence nature of selection, because a node

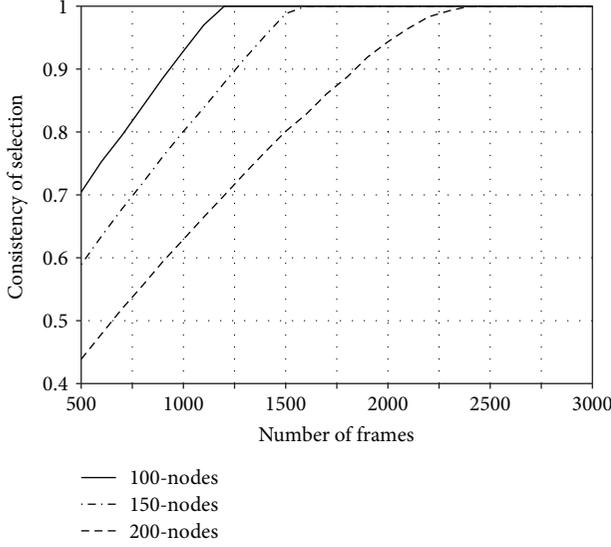


FIGURE 8: Convergence process of slot selection.

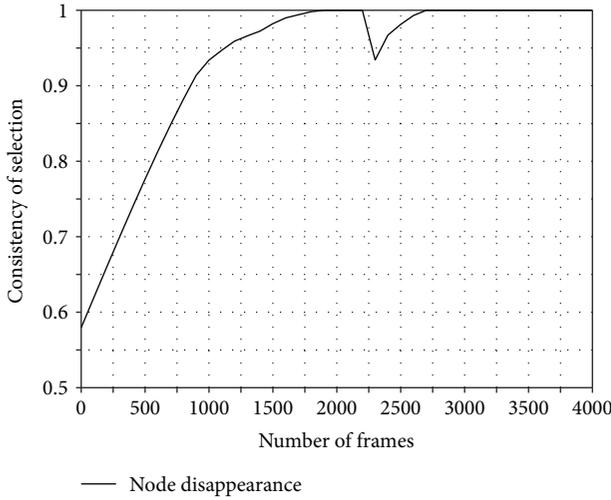


FIGURE 9: Reaction process of node disappearance.

always chooses the same slots after the convergence. According to these results, the consistency value with 100 nodes can reach its maximum at almost 1200th frame, and the convergence point is approximate at 1150th frames ahead with the consistency value of 200 nodes. It is obvious to discover that the climbing speed with 100 nodes is much faster than the climbing speed with more nodes. The direct reason of this phenomenon comes from the fact that the more nodes are involved in the network. The exploration problem becomes more complex, the corresponding convergence time should be longer than the scenario with less nodes.

To observe the adaptability of DSQ, the corresponding simulations are conducted. The reaction process of node disappearance is depicted in Figure 9. When the network condition is stable, the selection converges at the approximate 1800th frame. A node disappears at 2200th frame, even

TABLE 2: Impact of the learning rate α .

Metrics	α			
	0.1	0.15	0.20	0.25
t_c	860	690	620	590
V_f	0.72	0.76	0.83	0.88

TABLE 3: Impact of shrinking factor ρ .

Metrics	ρ			
	2	4	6	8
t_c	910	680	650	630
V_f	0.78	0.79	0.87	0.91

though consistency value drops a little bit, it quickly recovers to the maximum value at almost 2600th frame. Thanks to the learning memory, the climbing speed is much faster than the speed of the first convergence starting with the same level of consistency. The situation of appearing a new node is similar to this reaction process. After a period of exploration, the value of selection consistency will return to its maximum.

There are two metrics to evaluate the impact of learning rate α on the performance of Q-learning. The first one is the convergence time t_c which is denoted by the number of frames. Meanwhile, the quality of final solution is another significant metric, which is decided by the optimization objective in 1, and it can be evaluated by using a simple linear fitness function as follows:

$$V_f = \frac{1}{3} \sum_{i=1}^3 \frac{v_i - v_i^{\min}}{v_i^{\max} - v_i^{\min}}. \quad (12)$$

where V_f is the final fitness value, and v_i is the evaluation value of i th objective. The simulation results can be found in Table 2. As the increase of α , the convergence time is obviously decreased due to the acceleration of learning rate. When α reaches 0.25, the convergence time is reduced to 69%. However, the quality of solution is concurrently decreased along with this tendency; the fitness value with the maximum α becomes 1.2 times more than the value with the minimum α . The possible reason is caused by prematurity of exploration, where a node quickly falls into an unsatisfactory solution.

Another important parameter is the shrinking factor ρ , which controls the randomness of selection. As the augment of ρ , the random searching probability is distinctly decreased, and the corresponding result can be found in Table 3. When $\rho = 4$, the solution keeps at the good position, and the convergence time is also acceptable, so it can be considered as the best option among these values. From $\rho = 2$ to $\rho = 4$, the convergence time is decreased by about 25%. Even though the convergence time can be reduced by increasing ρ , the quality of scheduling solution becomes worse, because the scheduling quickly falls into a local optimal solution and loses the capability to jump out. These two metrics could be

conflicting in many cases; therefore, the best solution can be viewed as the best trade-off between them.

6. Conclusion and Future Work

In this paper, the concept of data aggregation is studied into the time slot scheduling of WSNs. Alongside with transmission delay, energy consumption also becomes an important performance index. Before explaining the details of the proposed approach, the concrete problem of discovering the optimal slot scheduling is analyzed and defined. Subsequently, the distributed implementation of aggregation scheduling policy based on Q-learning is described, the selection of time slot is abstracted as a Markov decision process. Thanks to the self-learning feature, the scheduling sequence automatically converges to a near-optimal sequence after a short period of exploration. The corresponding simulations are conducted by comparing DSQ with other three common WSN scheduling policies, and the results are valuable for thoroughly understand the performance of DSQ, when compared with other state-of-the-art approaches.

Although the simulation platform can evaluate the theoretical performance of the proposed scheduling policy in different application scenarios, the simulation results are still not the same as the measurements from the real devices in some cases. Therefore, the next step of work is to implement this method on the real sensor nodes, then observe and analyze the actual performance on these devices. Besides, the number of upstream nodes, which is closely related to the routing structure, is an important parameter in the learning process of this slot scheduling policy. The change of routing structure may directly impact the scheduling results. In the next step of work, some specific techniques should be developed to make the learning process adapt to the dynamic changes of routing structure.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported in part by the National Natural Science Foundation of China (NSFC: 71761007) and the project of the innovation team of knowledge engineering-based manufacturing service in the colleges and universities of Guizhou ([2015]58) and Young Science and Technology Talents Growth Project of Guizhou Provincial Education Department ([2017]124).

References

- [1] M. Rajiullah and S. Shimamoto, "An energy-aware periodical data gathering protocol using deterministic clustering in wireless sensor networks (WSN)," in *2007 IEEE Wireless Communications and Networking Conference*, pp. 3014–3018, Kowloon, China, March 2007.
- [2] T. Pham, E. J. Kimand, and M. Moh, "On data aggregation quality and energy efficiency of wireless sensor network protocols - extended summary," in *First International Conference on Broadband Network*, pp. 730–732, San Jose, CA, USA, October 2004.
- [3] J. Lin, N. Xiong, A. V. Vasilakos, G. Chen, and W. Guo, "Evolutionary game-based data aggregation model for wireless sensor networks," *IET Communications*, vol. 5, no. 12, pp. 1691–1697, 2011.
- [4] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, 2007.
- [5] Y. Lu, I. S. Comsa, P. Kuonen, and B. Hirsbrunner, "Adaptive data aggregation with probabilistic routing in wireless sensor networks," *Wireless Networks*, vol. 22, no. 8, pp. 2485–2499, 2016.
- [6] A. Bachir, M. Dohler, T. Wetteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [7] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, "Aggregation convergecast scheduling in wireless sensor networks," *Wireless Networks*, vol. 17, no. 2, pp. 319–335, 2011.
- [8] Y. Bo, L. J., and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pp. 2159–2167, Rio de Janeiro, Brazil, April 2009.
- [9] I. S. Comsa, M. Aydin, S. Zhang, P. Kuonen, J.-F. Wagen, and L. Yao, "Scheduling policies based on dynamic throughput and fairness tradeoff control in LTE-A networks," in *39th Annual IEEE Conference on Local Computer Networks*, pp. 418–421, Edmonton, AB, Canada, September 2014.
- [10] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregation scheduling algorithms in wireless sensor networks: solutions and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1339–1368, 2014.
- [11] P. Suriyachai, U. Roedig, and A. Scott, "A survey of MAC protocols for mission-critical applications in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 240–264, 2012.
- [12] J. Niu and Z. Deng, "Distributed self-learning scheduling approach for wireless sensor network," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1276–1286, 2013.
- [13] A. Somov, I. Minakov, A. Simalatsar, G. Fontana, and R. Passerone, "A methodology for power consumption evaluation of wireless sensor networks," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–8, Mallorca, Spain, September 2009.
- [14] Y. Chu, S. Kosunalp, P. D. Mitchell, D. Grace, and T. Clarke, "Application of reinforcement learning to medium access control for wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 23–32, 2015.
- [15] P. Wang, Y. He, and L. Huang, "Approaching the optimal schedule for data aggregation in wireless sensor networks," in *Wireless Algorithms, Systems, and Applications. WASA 2010. Lecture Notes in Computer Science*, G. Pandurangan, V. S. Anil Kumar, G. Ming, Y. Liu, and Y. Li, Eds., vol. 6221, Springer, Berlin, Heidelberg, 2010.
- [16] S. C. H. Huang, P. J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 366–372, Barcelona, Spain, May 2007.
- [17] B. Kang, P. K. H. Nguyen, V. Zalyubovskiy, and H. Choo, "A distributed delay-efficient data aggregation scheduling for

- duty-cycled WSNs,” *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3422–3437, 2017.
- [18] X. Chen, X. Hu, and J. ZhuX. Jia, J. Wu, and Y. He, “Minimum data aggregation time problem in wireless sensor networks,” in *Mobile Ad-hoc and Sensor Networks. MSN 2005. Lecture Notes in Computer Science*, vol. 3794, pp. 133–142, Springer, Berlin, Heidelberg, 2005.
- [19] Y. Lu, P. Kuonen, B. Hirsbrunner, and M. Lin, “Benefits of data aggregation on energy consumption in wireless sensor networks,” *IET Communications*, vol. 11, no. 8, pp. 1216–1223, 2017.
- [20] A. Kopke, M. Swigulski, and K. Wessel, “Simulating wireless and mobile networks in OMNeT++ the MiXiM vision,” in *Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks and Systems*, pp. 206–207, Marseille, France, March 2008.
- [21] M. Lee and V. W. S. Wong, “An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks,” in *PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005*, pp. 300–303, Victoria, BC, Canada, August 2005.



Hindawi

Submit your manuscripts at
www.hindawi.com

