IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 18, 2016, accepted October 26, 2016, date of current version November 18, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2622720

Distributed Degree-Based Link Scheduling for Collision Avoidance in Wireless Sensor Networks

BYUNGSEOK KANG, SUNGHO MYOUNG, AND HYUNSEUNG CHOO
College of Information and Communication Engineering, Sungkyunkwan University, Seoul 440-746, South Korea

Corresponding author: H. Choo (choo@skku.edu)

This work was supported in part by the Ministry of Science, ICT and Future Planning, South Korea, Institute for Information and Communications Technology Promotion through the G-ITRC Program under Grant IITP-2015-R6812-15-0001 and in part by the National Research Foundation of Korea within the Ministry of Education, Science and Technology through the Priority Research Centers Program under Grant 2010-0020210.

ABSTRACT Wireless sensor networks (WSNs) consist of multiple sensor nodes, which communicate with each other under the constrained energy resources. Retransmissions caused by collision and interference during the communication among sensor nodes increase overall network delay. Since the network delay increases as the node's waiting time increases, the network performance is reduced. Thus, the link scheduling scheme is needed to communicate without collision and interference. In the distributed WSNs environment, a sensor node has limited information about its neighboring nodes. Therefore, a comprehensive link scheduling scheme is required for distributed WSNs. Many schemes in the literature prevent collision and interference through time division multiple access (TDMA) protocol. However, considering the collision and interference in TDMA-based schedule increases the delay time and decreases the communication efficiency. This paper proposes the distributed degree-based link scheduling (DDLS) scheme, based on the TDMA. The DDLS scheme achieves the link scheduling more efficiently than the existing schemes and has the low delay and the duty cycle in the distributed environment. Communication between sensor nodes in the proposed DDLS schemes is based on collision avoidance maximal independent link set, which enables to assign collision-free timeslots to sensor nodes, and meanwhile decreases the number of timeslots needed and has low delay time and the duty cycle. Simulation results show that the proposed DDLS scheme reduces the scheduling length by average 81%, the transmission delay by 82%, and duty cycle by over 85% in comparison with distributed collision-free low-latency scheduling scheme.

INDEX TERMS Link scheduling, collision avoidance, TDMA, degree-based link scheduling, distributed wireless sensor networks, scheduling length, low latency, low duty cycle.

I. INTRODUCTION

Wireless Sensor Networks (WSN) are sensitive towards factors like transmission delay and energy efficiency, which are based on communication between sensor nodes, and these factors affect the overall performance of the WSNs [1], [28]. In contrast to centralized algorithms for WSNs, distributed algorithms [29], [30] have limited information of neighbor nodes or a particular region, which makes it difficult for them to achieve the high performance. To attain efficient communication between the nodes, interference must be considered. Increased retransmissions caused by interference and collision, significantly reduces the life time of the sensor nodes [2]. To avoid collisions in communication between neighbor nodes, two types of interferences are to be considered. First one is primary interference where a node receives transmissions from multiple neighbor nodes at the same instant. The other is secondary interference, and it occurs when a communication between a pair of nodes unintentionally interfere with the communication of neighboring pair of nodes [3].

There have been two approaches studied for the Medium Access Control (MAC) in WSN. Some suggest an 802.11 like CSMA based MAC algorithms in WSN. CSMA based MAC is also known as contention based MAC, where nodes contend to gain access of the channel. Such algorithms require nodes to constantly listen to the channel and increase the number of transmissions due to collisions. Therefore CSMA based MAC algorithms require more energy and hence decreases the life time of a node [1], [4], [5]. To resolve this issue many turn toward TDMA based MAC algorithms which are contention free algorithms [6]–[8], however they require a scheduling algorithm at the beginning to assign the timeslots. TDMA constructs a frame by using scheduling algorithms. Frame consists of equally sized timeslots and these timeslots are assigned to nodes/links for transmission. TDMA based scheduling algorithms can be generally categorized into two type: 1) where nodes are assigned timeslots [9], [10], and 2) where links between nodes are assigned timeslots, known as link scheduling [11]–[13].

TDMA based link scheduling has received great attention over the past few years. Weizhao et al in [12] presents a centralized and distributed TDMA based link scheduling algorithm, which considers more realistic network models, and their centralized algorithm is near optimum and its distributed implementation is not much far away. S. Gandham et al in [11] presents a distributed TDMA based link scheduling algorithm. They distribute their algorithm in two phases. In first phase they assign color to each link in the network in a way that two links which are connected to same node are assigned different colors, and achieves to color the network in at most $(\delta + 1)$ colors. In the second phase they assign unique timeslot to each link based on the color, and also took direction of transmission into account as well which helped to avoid hidden and exposed terminal problem.

Distributed collision-free Low-latency Scheduling (DCLS) scheme provides link scheduling in WSN based on improved TDMA. DCLS scheme saves the energy by applying low duty cycle and uses the graph coloring theory to have low latency without the packet collision [13]. Collisions in DCLS scheme are avoided through the use of strong edge coloring theory. Strong edge coloring does not choose the same color from one edge to the adjacent two edges in two-hop domain [14]. This means that for the link scheduling different timeslots are assigned to prevent the packet collision on the continuously connected three links in TDMA protocol. However, since the graph coloring theory can color an edge with a single color, therefore it assigns single timeslot to each link [15]. To improve above feature, DSCL scheme assigns the several timeslots to a link thereby it tries to reduce delay time. However, some links could not be assigned with timeslots to avoid the collision, whereas some links are assigned with several timeslots. Accordingly, the scheduling length increases because the links that do not have timeslots have the decreased opportunity to assign with the timeslots. Moreover, even though the links communicated without the collision in a timeslot, the overall bandwidth decreases because some links are not assigned with timeslots.

Shortcomings like increased schedule length and inefficient bandwidth utilization of the DCLS scheme are mitigated by proposed Distributed Degree-based Link Scheduling (DDLS) scheme which is presented in this paper. In the proposed DDLS scheme each node collects the degree information from its one hop neighbors, where degree is defined as number of one hop neighbors of a node. After collecting the degree information a node takes the sum of all the collected degrees including its own. DDLS scheme follow the sequential heuristic of ''Largest sum of degree first'' [22]. This means that the node with the largest value after summation of degrees will get the opportunity first to make a transmission pair. Largest sum of degree of any node effectively means that the node is in the high density region. DDLS scheme assigns a time slot to multiple links in a way that primary and secondary interferences are avoided, and this distribution of time slot assignment is even throughout the schedule length. Also one link is only assigned one time slot which increases the probability of scheduling for other nodes. In contrast DCLS scheme assigns multiple time slots to one link and number of links assigned per time slot are unevenly distributed. These factors in DDLS scheme reduce the transmission latency and schedule length. More detailed explanation of proposed DDLS scheme is presented in the section IV. Simulation results show that based on the number of nodes the proposed DDLS scheme reduces the schedule length by maximum 66%, the transmission delay by maximum 88%, and network activity time by 80% in comparison to the DCLS scheme. Reduction in schedule length, transmission delay and network average activity improves the energy consumption of the WSNs and hence increase the network life tim. This work is extended version of our previous work [27] which published International Conference on Ubiquitous Information Management and Communication (ICUIMC) 2014.

Rest of the paper is structured in the following manner. Section 2 presents the background and related work concerning link scheduling in WSNs. Network model and in depth analysis of the DCLS scheme in terms of performance improvement opportunities is presented in section III. Background, methodology and functioning of the proposed DDLS scheme to reduce the schedule length, transmission delay and energy consumption is described in section IV. Towards the end section V discusses the simulation based performance evaluation. Section 6 concludes the paper by presenting conclusion and future works.

II. RELATED WORK

Many existing link scheduling schemes uses the TDMA protocol and the graph coloring theory [9], [11], [16], [17] for the collision free communication between sensor nodes. In particular, the graph coloring theory avoids the collision among nodes by utilizing the independent set and achieves speedy link schedule [18], [19].

In an independent set two neighboring nodes cannot have the same color [20]. **Fig. 1** explains the independent set by pivoting on the black node. The first Not Independent Set cannot be the independent set since the black nodes are neighbor to each other. Second case is an Independent set as the black nodes are not in the neighborhood, but there exist a possibility for another black node. The third Maximal Independent Set has the two of black nodes, but there is no possibility for any other node to be black, in comparison with the second Independent Set. When no more black node can be made during the creation process of Independent Set in a

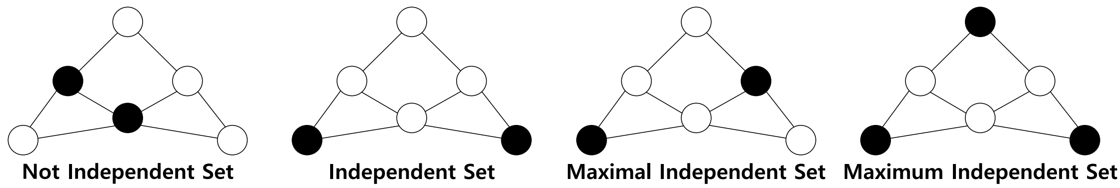


FIGURE 1. Case of independent sets.

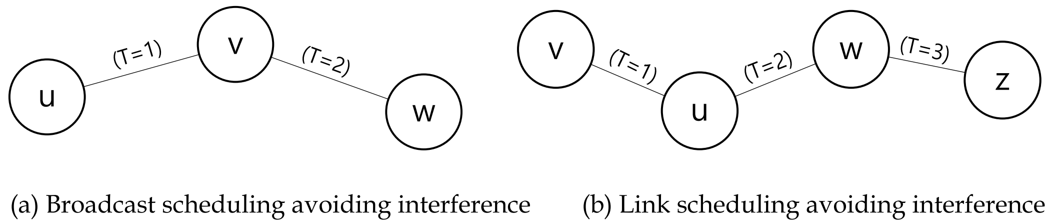


FIGURE 2. Broadcast scheduling and link scheduling protocols. (a) Broadcast scheduling avoiding interference. (b) Link scheduling avoiding interference.

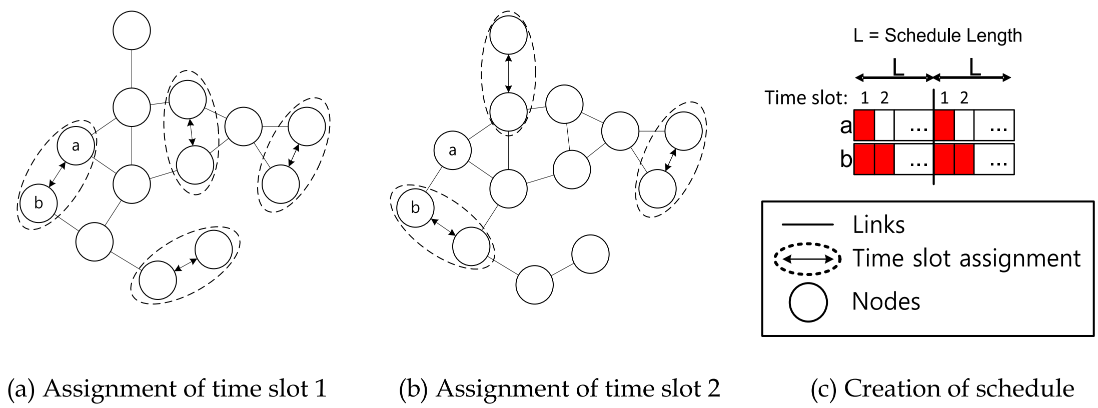


FIGURE 3. Examples of time slots assignment in link scheduling. (a) Assignment of time slot 1. (b) Assignment of time slot 2. (c) Creation of schedule.

graph, the graph becomes Maximal Independent Set. Lastly, the fourth Maximum Independent Set consists of the three of black nodes in comparison with other examples. It has the most number of black nodes. In particular, to find the independent set in the distributed wireless sensor networks is the NP hard problem [21], but the link scheduling can be quickly completed if the links are assigned with timeslots by utilizing Maximal Independent Set close to Maximum Independent Set. Section 4.3 explains the proposed link scheduling method which utilizes the maximal independent set.

To avoid collision, various link scheduling schemes use the TDMA protocol. Two variations of TDMA protocol are the broadcast scheduling protocol and link scheduling protocol [6]. As an example, the **Fig. 2** presents the two types of scheduling in TDMA protocol. In the broadcast scheduling protocol as shown in the **Fig. 2(a)**, two hop nodes are assigned with different time slots (T). If node u communicates with node v, and node w communicates with node v, the packet collision occurs because node v receives the packets from node v and node w (i.e. Primary interference). Thus, the packet collision can be avoided by assigning the different timeslots to the links between node *u* and *v* and node *v* and *w*. In the link scheduling protocol as shown in the **Fig. 2(b)**, three continuous links are assigned with the different timeslots. If node *u* communicates with node *v*, and node *w* communicates with node *z*, the packet of node *v* conflicts with the packet of node *w* because node *u* is in the communication radius of node *w* (i.e. Secondary interference). Thus, to avoid the packet collision, node *u* waits until the communication between node *w* and node *z* is closed. Therefore, the three links are assigned with the different timeslots to prevent the collision and interference in the link scheduling.

The **Fig. 3** presents the link scheduling based on basic TDMA in WSNs. **Fig. 3(a)** and **3(b)** shows the first and second timeslot allocation for the link. In **Fig. 3(a)**, timeslot 1 is assigned to the node *a* and *b*, and in the **Fig. 3(b)**. Node *b* and another node that is not node *a*, are assigned timeslot 2.

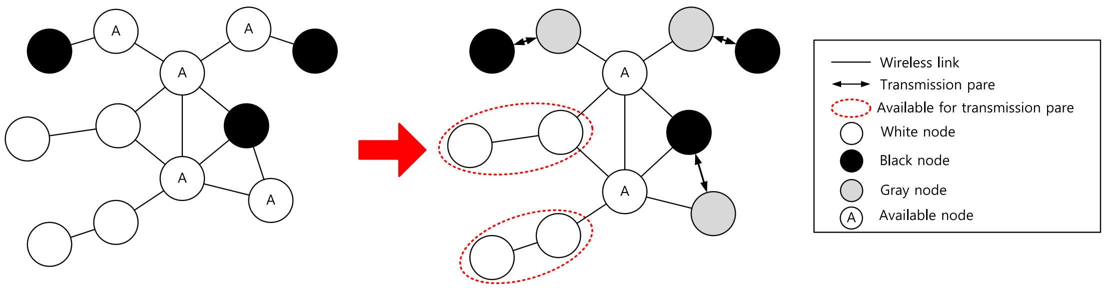


FIGURE 4. DCLS scheme with a small number of black nodes.

Node b communicates with the neighbor node, but the collision and interference do not occur because the communication between node b and the neighbor node uses the different time as shown in the **Fig. 3(c).** The four links in the **Fig. 3(a)** indicates timeslot 1 and the three of links in the **Fig. 3(b)** uses timeslot 2 and communicate without any collision.

The DCLS scheme colors each node by using the graph coloring theory based on the TDMA protocol. DCLS scheme classify nodes into white, black, and gray nodes. Initially all the nodes in the network are white nodes. The black node is the one which is included in the independent set, which is created randomly from the white nodes. The nodes selected as the black node have the right to make the link with the neighbor nodes. Lastly, the gray node is the neighbor of a black node. If multiple gray nodes exist as the neighbors of a black node, one of the gray node is randomly selected and transmission link is assigned with the timeslot. The algorithm performs above process repeatedly until all links are assigned with timeslots.

III. PRELIMINARIES

This paper assumes the network model as follows. n fixed sensor nodes are randomly deployed in the field. All the sensor nodes have the forwarding antenna, and the communication range 'r' is uniformly distributed. The network communication graph is descripted as $G = (V, E)$. $V = \{v_1, v_2, \ldots, v_n$ is a set of the nodes, and E is a set of all the edges. If $\{v_i, v_j\} \subseteq V$, Edge $e = (v_1, v_2) \in E$. The model performs the link scheduling by considering the primary and secondary interference (explained in Section 2) during the communication.

Algorithm in the basic DCLS scheme depends heavily on random factors. The time to finish the link scheduling, the network delay time, and the energy efficiency are not constant [31], [32]. Also, since the DCLS algorithm does not consider the link assigned with timeslot already and reassigns the timeslots, the efficiency of the link scheduling deceases [33]. The factors involved in decreasing the performance in the basic DCLS scheme are the number of black node, positioning, selection of available nodes, and selection of gray node.

The **Fig. 4** shows that the number of black nodes selected as the minimal independent set. The black node has the right to make the transmission pair and selects one of the neighbor nodes for making the pair. The first graph shows the selection of three black nodes as the independent serve set from 12 of white nodes. If the node connects with a black node, it becomes an available node. Since node v connects with two of the black nodes, it cannot be the available node. An available node can be a gray node and makes a transmission pair with its neighboring black node. Around three black nodes, there are four of nodes that can be the available node as a gray node. The second graph shows that black node z choose randomly node w as a gray node between the available node u and w. After finishing the selection, the link connecting the two nodes is assigned the timeslot. Although the primary and secondary interference are preventing by above method, however there are still links available in the graph to whom timeslot could have been assigned without primary and secondary collision. These are mentioned in the second graph of **Fig. 4** through dotted ellipse. The reason why these links cannot be assigned with the timeslot is because there is no node selected as the first black node. Thus the links that could communicate in the same timeslot are wasted. It causes the efficiency of transmission bandwidth decreases, and the scheduling length increases.

The **Fig. 5** shows the case that is contrary to the **Fig. 4**. Now there are more black nodes. Since a lot of the black nodes formed as the independent set, there is only one node as an available node among the neighbors. If the block nodes are formed as Maximal or Maximum Independent Set, the number of available nodes that can be the gray node decreases. Moreover, since the prescribed independent set cannot change, the number of black nodes and its positions cannot be altered. Therefore, the only one link can be assigned with the timeslot. Since the timeslots are wasted in this case, the scheduling length increases.

While performing the link scheduling, if multiple links are assigned with the same timeslot without any collision, then it is possible to perform the link scheduling quickly. The graph (a) and (b) in the **Fig. 6** has five of the black nodes formed as the independent set. Since each black node in the graph (a) is placed on the proper place, all of the black nodes

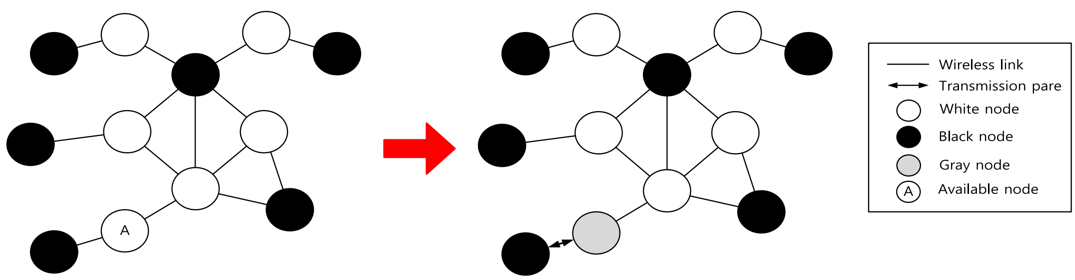


FIGURE 5. DCLS scheme with a large number of black nodes.

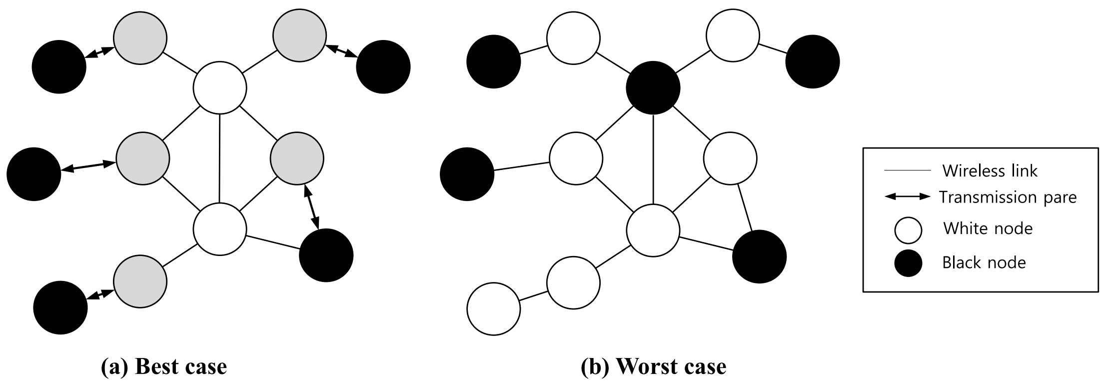


FIGURE 6. Best case and worst Case in DCLS scheme. (a) Best case. (b) Worst case.

make transmission pair with the neighboring grey node and transmission links are assigned with the same timeslot. In the graph (b), the gray node cannot be generated because of the position of the black nodes. This shows the worst result since no links can be generated and timeslot cannot be assigned. As a result, the DCLS scheme performs differently depending upon the situation. If the link scheduling scheme provides the stable performance without the collision and that all of links are assigned with the timeslots, the scheduling length and the delay time decrease, and the energy can be saved. We describe the proposed DDLS link scheduling scheme in section IV.

IV. DISTRIBUTED DEGREE-BASED LINK SCHEDULING SCHEME

In this section, detail description of the proposed collision and interference free Distributed Degree-based Link Scheduling (DDLS) scheme with low latency and low duty-cycle, is presented. In DCLS scheme, even under same network environment schedule length shows inconsistent behavior because of random factors. If the number of required timeslots increases for schedule completion, it increases the waiting time for the sensor nodes and inevitably consume more energy. Also, DCLS scheme experiences the number of wasted time slots which also causes the network latency to be increased. Proposed DDLS scheme reduces the schedule length by assigning a time slot to multiple available links without collision, based on Collision Avoidance Maximal Independent Link Set (CAMILS). CAMILS is effectively a MIS of the links with collision avoidance. This provides an opportunity for multiple links to be assigned a same time slot without collision, which enhance the bandwidth utilization efficiency and reduce the wait period of the sensor for the transmission, which infect improves the life time of the network. With the fixed upper bound for the time to assign the time slots, the proposed DDLS scheme provides better link scheduling results comparing to the DCLS scheme in terms of transmission delay. In the DDLS scheme, at any given instant each node is in one of the following states: READY, WAIT, PAIRED, BLOCK and COMPLETE. Change in the state of the node occurs when a message is received internally or from some other node. All the links attached to a node are scheduled when it reaches the COMPLETE state. It is being assumed that all the nodes in V are synchronized in terms of time. In this paper we introduce a time duration called 'work period', in which each node tries to make a transmission pair with one of its neighbor nodes and assign a timeslot to the link between them according to the DDLS scheme. At the end of the work period each node updates the information and gets back to the READY state and revitalizes its efforts to make a transmission pair on one of its unpaired links. Duration of work period is considered to be long enough for a message to be transmitted across two hop neighbors. A node can have a bidirectional communication in one timeslot. Notations used

TABLE 1. Notation.

Variables	
id_i	It represents the ID of node i where $i \in V$
d_i	It represents the number of neighbors of node i where $i \in V$, and is called degree of the node
ds_i	It represents the sum of degrees value at node i where $i \in V$, which is calculated by using the eq.1. Initially ds_i equals to d_i
Sets	
V	It represents all nodes in the network
E	It represents all links in the network
$N(v)$	It represents neighbors of node v
$NS(v)$	It represents the "sum of degrees" of neighbors of node v, in the form of (id_u, ds_u), where $u \in N(v)$

to describe the functioning of DDLS scheme are mentioned in the table 1.

A set V presents all the nodes in the WSN, and every node in V is referred as node v. As DDLS is a distributed scheme therefore DDLS scheme runs on each node in V at the same time. For better description of DDLS scheme we consider that node v represents one node, as shown in **Fig. 7(a)**, and all the neighbors of node v are referred as node u where $u \in N(v) \cap V$. In the proposed DDLS scheme initially each node v has its degree information d_v. Each node v broadcasts d_v to its neighbors. When a node receives d_u from its neighbor where $u \in N(v)$, it also contains the information of its two hop neighbor. Through this method sufficient information of three continuous links is obtained which is then utilized by DDLS scheme to determine the collision and interference free schedule for the node v.

In DDLS scheme timeslot assignment to a link primarily depends on the 'sum of degree' (ds_v) information at the incident node v. ds_v is a sum of the degree value of node v (d_v) with degree values of all its neighbor nodes (d_u where $u \in N(v)$). As a prerequisite of DDLS scheme, each node v collects the degree (d_u where $u \in N(v)$) of its neighbors and calculates the ds_v using the following equation.

$$ds_v = \sum_{u \in N(v) \cup v} d_u \tag{1}$$

Each node v sends its ds_v to all neighbors, and collects their ds_u (where $u \in N(v)$) and maintains them in the local table $NS(v)$ as a tuple of (id_u, ds_u). **Fig. 7(a)** shows the degree value of the nodes in a WSN and that how this information is shared among neighbors. **Fig. 7(b)** shows the calculated ds_v value of each node using the Eq (1). The probability of the collision and interference depends on the node v sum of degree; therefore the node with higher sum of degree value has more probability for collision and interference. In **Fig. 7(b)** the node v has the largest sum of degree value considering its two hop neighbors; therefore its probability for collision and interference is the highest. Also it is worth noticing that while calculating sum of degree one hop links are considered two times, once as a link of node v and other time link of the one hop neighbor. Whereas two hop link information is considered only once in the sum of degree value. This implies that sum of degree value considers primary interference (collision) more than secondary interference. To avoid the collision and interference three continuous links are assigned different time slots based on sum of degree value. Also it is well known that introducing some heuristic strategy improves the performance of the distributed algorithm. E.g. Authors in [22] shows that using Largest First, or Smallest Last order improves the performance of the distributed algorithm. Therefore, based on above reasoning, DDLS scheme uses the sum of degree value of nodes to assign the time slots to the links.

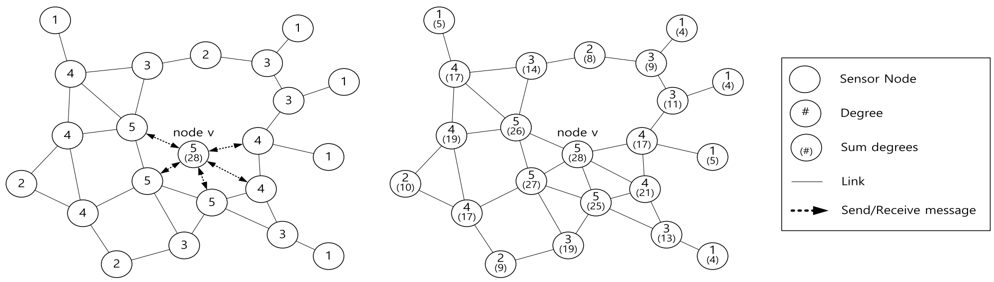
DDLS is a distributed scheme in which all the sensor nodes run the scheme simultaneously to assign the timeslots to their links. While running the DDLS scheme, at any given instant each node is in one of the following states: READY, WAIT, PAIRED, BLOCKED and COMPLETE. Transition between the states occurs when a node receives a message internally or from its neighbor node. Internal message is defined as a message sent to the node by itself as a result of an event occurred within the node. DDLS scheme initiates when a node v enters in the READY state after receiving the ds_u (where $u \in N(v)$) from all its neighbors. A node transits to the COMPLETE state when all the links attached to it are assigned timeslots. **Fig. 8** shows the state transition in the node v according to proposed DDLS scheme as a reaction to the messages within one work period. The details of the states and their transitions are described below.

A. READY STATE

In DDLS scheme a node v with the largest ds_v value among its neighbors gets the opportunity to assign a timeslot to a link which node v shares with a neighbor node with largest ds_u from $NS(v)$. Every time node v enters in the READY state it must compare its ds_v value with ds_u values maintained in $NS(v)$ to determine if node v has the opportunity to assign the timeslot. This functionality is implemented through an internal message (*internal_msg_contend*). On receiving *internal_msg_contend* node v performs the comparison between its ds_v and ds_u (where $u = MAX(NS(v))$), If ds_v is greater than ds_u, then node $v$sends a *join_req_msg* to node u and moves to the WAIT state. $MAX(NS(v))$ is defined as a function which returns a node u with maximum ds_u value. $MAX(NS(V)) \rightarrow u$ where $u \in NS(V)$ and $u \gg x$ for any node $x \in NS(V)$. An operator $\gg$ has been used to present multiple comparisons between two nodes and can be defined as:

$$x \gg y \text{ is true if } ds_x > ds_y \text{ or if } ds_x = ds_y \text{ and } id_x > id_y$$

On receiving *join_req_msg* from a neighbor node u, node v checks whether u is equal to $MAX(NS(v))$ or not. If it is, then node v performs the make_pair_procedure and changes its state to PAIRED. If v receives a *join_req_msg* from some other node then $MAX(NS(v))$ then it ignores the request and stay in the READY state. On receiving the *primary_join_msg* from a neighbor node u, node v performs the primary_update_procedure and changes its state to the BLOCKED. Whereas on receiving the



(a) Request/Receive degree information (b) Nodes after calculating sum of degree

FIGURE 7. DDLS scheme: calculation of sum of degree value. (a) Request/Receive degree information. (b) Nodes after calculating sum of degree.

secondary_join_msg from a neighbor node u, node v performs the secondary_update_procedure and stays in the READY state. At the end of each work period, every node sends an internal message (internal_msg_workperiod) to itself. On receiving the message each node changes its state to READY from other state but except the COMPLETE state.

B. COMPLETE STATE

When node v has assigned timeslots to all its links the ds_v values becomes zero, and node v transit from its current state to COMPLETE state. In the COMPLETE state a node ignores all the received messages.

C. WAIT STATE

Node v transits to the WAIT state when it sends a *join_req_msg* to node u in READY state. Upon receiving the *join_accept_msg* from node u in the WAIT state node v performs the make_pair_procedure and transit to the PAIRED state. In case where node v receives a *primary_join_msg* from any neighbor 'i' (where $i \in N(v)$) while waiting for the *join_accept_msg* from the node u it implies that i has assigned a timeslot to one of its links and for avoiding the collision node v refrains itself from assigning the same timeslot to its link and thus performs the primary_update_procedure and transits to the BLOCKED state. In case where node v receives a *secondary_join_msg* from node u, node v performs the seconday_update_procedure and transits to the READY state. In other case where secondary_join_msg is received from a neighbor other than the node u, then node v simply performs the secondary_update_procedure and stay in the WAIT state.

D. PAIRED STATE

Node v transits to the PAIRED state only when it makes a pair with one of its neighbor nodes, by assigning a timeslot to the link between node v and the neighbor node. In the PAIRED state node v checks the ds_v value and if it is zero then it implies that timeslots are assigned to all the links attached to the node v, and in this case node v transits to the COMPLETE state. Node v cannot receive *primary_join_msg* or *join_req_msg* from its neighbor nodes, as to avoid collision all neighbor nodes transit to BLOCKED state when node v transits to the PAIRED state after making the pair. In the PAIRED state node v can receive only *secondary_join_msg*, and in that case it will perform the secondary_update_procedure and stay in the PAIRED state. From the PAIRED state, node v will transit to the READY state at the end of work period through internal_msg_workperiod.

E. BLOCKED STATE

To avoid the collision and interference node v transits to the BLOCKED state when it receives the *primary_join_msg* in the READY or WAIT state form any neighbor node, because only in the READY and WAIT states a node v has possibility to make a transmission pair with the neighbor node. When transited to the BLOCKED state node v checks whether the ds_v is zero or not and in case where ds_v value is zero, node v will transit to the COMPLETE state. In the BLOCKED state, node v can receive *primary_join_msg* and *secondary_join_msg* from its neighbors, in case of both these messages node v performs the concerned update procedures and stay in the BLOCKED state. Node v transits to the READY state from the BLOCKED state, in the case of internal_msg_workeperiod at the end of work period.

Above explained functioning of the DDLS scheme is also presented in the Algorithm 1 with complete detail. Algorithm 1 describes that how DDLS scheme keeps running on a node until the node reaches the COMPLETE state. The node can receive external or internal message. Different functionalities are performed based on the node's current state and received message type.

When two nodes make a pair and assign a time slot to a link between them, then primary interference occurs if one hop neighbor also assigns the same time slot to one of its link. To avoid the primary interference a node which makes a pair, broadcasts a *primary join message* and a node which receives this *primary join message*, moves to a BLOCKED state for that work period. *Primary join message* also contains the updated sum of degree value of the sender node u, and the *primary join message* receiver node v updates this received ds_u in to the $NS(v)$ and also updates it sum of degree value, which is then broadcasted in the *secondary join message*. *Secondary join messages* are used to prevent the secondary interference. The receiver node of the *secondary join message* also updates the sum of degree value of the sender

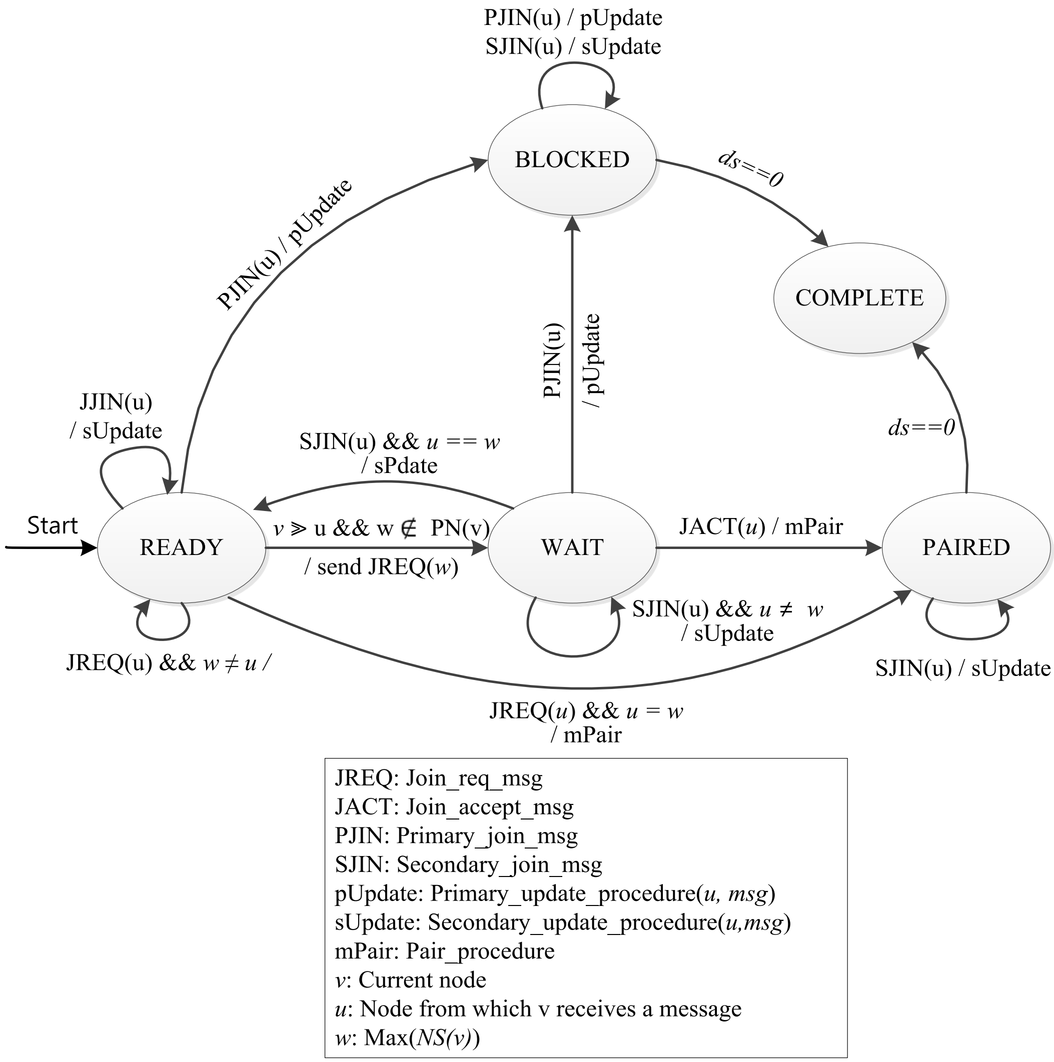


FIGURE 8. State diagram of node *v* with DDLS.

node in its $NS(v)$. Primary and secondary join messages together avoid the primary and secondary interference. This also effectively means that to avoid primary and secondary interference, three adjacent links are assigned different time slots. The nodes which transmit the primary and secondary join message do not participate in the process of making the pair in that particular work period, whereas this restriction doesn't apply on other nodes in the WSNs and they can participate in the process of making the pair in that work period. Algorithm 2, 3 and 4 explains the update of ds_v in case of make_pair_procedure, primary_update_procedure and secondary_update_procedure. **Fig. 9** describes the avoidance of primary and secondary interference by using the primary and secondary join messages.

1) MAKE_PAIR_PROCEDURE

In the DDLS scheme priority of node to assign a timeslot to a link is determined through the node's ds_v value which represents the number of links with unassigned timeslots, attached to a node and each of its neighbors. Nodes v and u must update their ds_v and ds_u values when a timeslot is assigned to a link between nodes v and u. As described earlier node's one hop links are considered twice while calculating the ds_v value, therefore when node v and u makes a transmission pair between themselves, they decrease their ds_v value by 2. This update of ds_v value is presented in the line 3 of algorithm 2. Further node v and u will broadcast the *primary_join_msg* which contains their updated ds_v and ds_u values. This make_pair_procedure is presented in the algorithm 2.

2) PRIMARY_UPDATE_PROCEDURE

Node v, on receiving the *primary_join_msg* from node u (where $u \in N(v)$) updates the $NS(v)$ with new ds_u value received in the message. Two hop links of node v are considered only once while calculating ds_v, therefore when a neighbor node's link (two hop link of node v) gets paired then node v decreases its ds_v by one. This is presented in the line 1 of algorithm 3. If a node v is a neighbor to both the nodes which are involved in making the pair then node v will receive

Algorithm 1 DDLS Algorithm for Scheduling Node v

```
Input: ds_v and NS(v), where v ∈ V
Output: Complete link schedule of node v
Initialize Time Slot (TS) with 0
u is the sender node where u ∈ N(v)
while (state != COMPLETE) {
  receive msg
  switch(state){
    case READY:
      if msg.type== internal_msg_contend && (v ≫ (w = MAX (NS (v)) )  &  &  w∉PN(v))
        send  join_req_msg(w) and set state = WAIT
      if msg.type== join_req_msg(u) && (u == (w = MAX (NS(v)))
          pair_procedure(u, msg), set state = PAIRED and send internal_msg_schedulecheck
      if msg.type == primary_join_msg(u)
          primary_update_procedure (u, msg)
        set state = BLOCKED and send internal_msg_schedulecheck
      if msg.type== secondary_join_msg(u)
        secondary_update_procedure (u, msg)
        set state = READY and send internal_msg_contend
      if msg.type == internal_msg_scheduleperiod
        set state = READY, increment TS by one and send internal_msg_contend
      break;
    case WAIT:
      if msg.type == join_accept_msg(u)
        pair_procedure(u, msg) , set state = PAIRED and send internal_msg_schedulecheck
          if msg.type == primary_join_msg(u)
        primary_update_procedure(u, msg)
        set state = BLOCKED and send internal_msg_schedulecheck
          if msg.type == secondary_join_msg(u) && u == MAX(NS(v))
            secondary_update_procedure(u, msg)
          set state = READY and send internal_msg_schedulecheck
      else
          secondary_update_procedure(u, msg) and set state = WAIT
      if msg.type == internal_msg_scheduleperiod
        set state = READY , increment TS by one and send internal_msg_contend
        break;
    case PAIRED:
      if msg.type == secondary_join_msg(u)
        secondary_update_procedure (u, msg) and set state = PAIRED
      if msg.type == internal_msg_scheduleperiod
        set state = READY , increment TS by one and send internal_msg_contend
      if msg.type== im_schedule_iscomplete
        if ds_v == 0
          set state = COMPLETE
      break;
    case BLOCKED:
      if msg.type == primary_join_msg(u)
        primary_update_procedure (u, msg)
        set state = BLOCKED and send internal_msg_schedulecheck
      if msg.type == secondary_join_msg(u)
        secondary_update_procedure (u, msg) and set state = BLOCKED
      if msg.type == internal_msg_scheduleperiod
        set state = READY , increment TS by one and send internal_msg_contend
      if msg.type== internal_msg_schedulecheck
        if ds_v == 0
          state = COMPLETE
        break;
    }
  }
```

Algorithm 2 Make_Pair_Procedure

Input: u and msg
Output: pair formation between v and u, *primary_join_msg*

```
if msg.type == join_req_msg
  send join_accept_msg
ds_v = ds_v − 2ds_v = ds_v−2
Pair formation by assigning current value of TimeSlot to the link between v and u
insert u into PN(v)
    insert ds_v in primary_join_msg primary_join_msg
broadcast primary_join_msg
```

Algorithm 3 Primary_Update_Procedure

Input: u and msg
Output: updated $NS(v)$ and *secondary_join_msg*

```
ds_v = ds_v − 1
update ds_u in NS(v) with the received ds_u value in primary_join_msg
insert ds_v in secondary_join_msg
broadcast secondary_join_msg
```

Algorithm 4 Secondary_Update_Procedure

Input: u and msg
Output: updated $NS(v)$

```
    update ds_u in NS(v) with the received ds_u value in secondary_join_msg
```

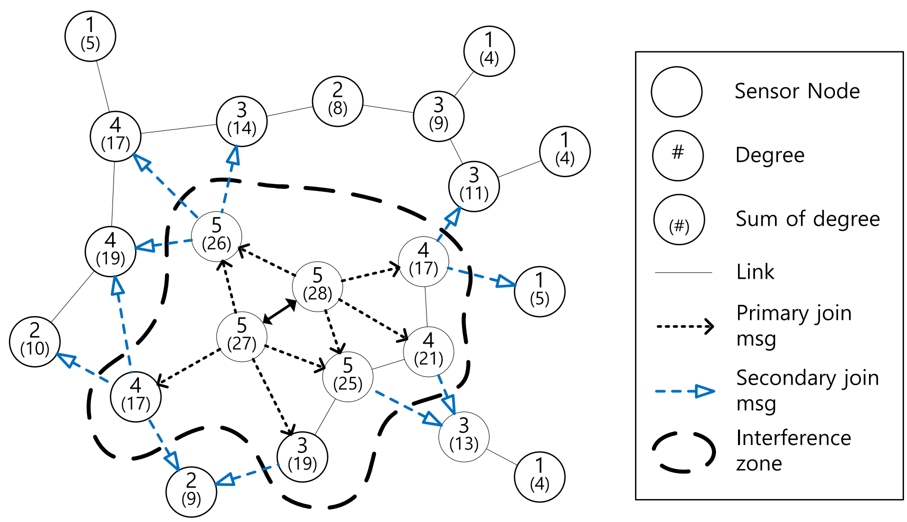


FIGURE 9. Primary and secondary interference avoidance.

primary_join_msg twice and will twice reduce its ds_v by 1. Further node v will broadcast the *secondary_join_msg* to its neighbors with the updated ds_v value. This ds_v update procedure based on *primary_join_msg* is described in algorithm 3 as primary_update_procedure.

3) SECONDARY_UPDATE_PROCEDURE

When a node v receives the *secondary_join_msg*, it extracts the ds_u and update the value in $NS(v)$, where u is the sender node. This procedure is presented as the secondary_update_procedure in algorithm 4. In the DDLS scheme node v maintains and update the $NS(v)$ with only neighbor ds_u where $u \in N(v)$; however ds_u also includes the information of two hop neighbor of node v, this effectively prevents both collision and interference while assignment of the timeslot to the link.

Fig. 10 presents the process of time slot assignment to all the links in DDLS scheme. In the **Fig.** 10 **(a)** and **(b)** the nodes make pair considering their sum of degree values, primary and secondary join message based on the procedure explained earlier in this section. **Fig. 10 (c)** presents the updated sum of degree values after making the pair and receiving the

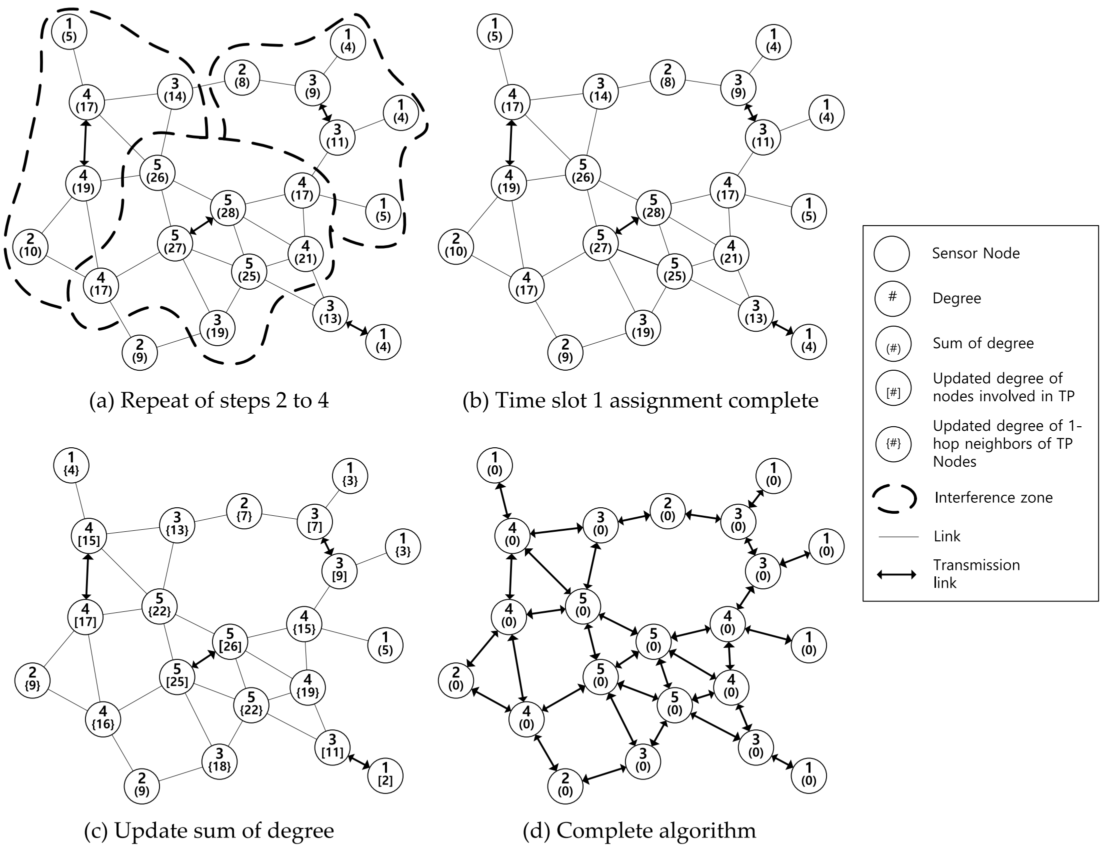


FIGURE 10. **Forming transmission pairs and updating sum of degree. (a) Repeat of steps 2 to 4. (b) Time slot 1 assignment complete. (c) Update sum of degree. (d) Complete algorithm.**

primary join message. Finally **Fig. 10 (d)** shows that on the completion of the algorithm the sum of degree value of all the nodes is zero and all the nodes have made pair with their neighbor node, which means that time slots are assigned to all the links.

The time complexity of the proposed DDLS scheme for the whole WSN is equal to L number of work periods where L is the total number of timeslots required to assign timeslot to each link and this is also a schedule length. This is because in the DDLS scheme a timeslot is assigned in every work period and therefore during the time algorithm runs on every node in the WSN no work period goes wasted. For a specific node v in the WSN the time complexity of the DDLS scheme can have maximum value of L, and this is when v is the last node in the WSN to assign the timeslot to its link. The minimum value for time complexity of the node v is d_v which is number of neighbor nodes of v, and this is when the ds_v value of the node v is maximum in the WSN. The maximum and minimum values of time complexity for node v are defined in the following formula.

$$T_v = \begin{cases} L & \text{if } ds_v = min_{u \in V}\,(ds_u) \\ d_v & \text{if } ds_v = max_{u \in V}\,(ds_u) \\ & otherwise \end{cases} \tag{2}$$

Message complexity is also an important parameter in WSN which can be defined as number of messages received and transmitted by the nodes in the WSN. A node sends a message when it makes a transmission pair with neighbor node, when it sends the primary join message to neighbor nodes after making the transmission pair and when it sends the secondary join message after receiving the primary join message. A node sends an individual message to the neighbor node to make a transmission pair or in response to the transmission pair request from the neighbor. In the case of primary and secondary join messages a node send one message which is broadcasted to all the neighbor nodes. From this transmitted message complexity of node v can be defined as:

$$v_{tx} = d_v + 2 \tag{3}$$

Similarly a node v receives a message when it makes a transmission pair, when it receives a primary join message and when it receives a secondary join message. Number of messages received by node v during the process of making transmission pairs, depend on the value of d_v. In case of primary and secondary join messages number of received messages on node v depends on the d_v value of the neighbor nodes and two hop neighbor nodes. Number of received messages by node v can be defined as:

$$v_{rx} = d_v + \sum_{u \in N(v)} d_u + \sum_{u \in N(v)} \sum_{i \in N(u)} d(i) \tag{4}$$

Based on the eq. 3 and 4 the number of messages transmitted and received for the whole WSN can be defined as follows.

$$Msg_{tx} = \sum_{v \in V} v_{tx} \tag{5}$$

$$Msg_{rx} = \sum_{v \in V} v_{rx} \tag{6}$$

V. PERFORMANCE EVALUATION

In this chapter we present simulation results of our proposed DDLS scheme, and compare the results of schedule length, latency, duty cycle (activation nodes ratio) with DCLS scheme under same simulation environment. The simulations consider two environments for objective comparison. First, simulations are performed under same environment as of DCLS scheme. In second environment, simulations are performed without limiting any variable. Two simulation environments use same network size of 200m x 200m. Sensor nodes are fully connected and in the initial step 20 nodes are created, which are then increased to 100 in nine steps by increasing 10 nodes in each step. Both simulation environments uses the same process of nodes increase as mentioned earlier. However variable of maximum degree behaves differently in the two simulation environments. In the first simulation environment value of maximum degree is initially six, and in each step it is increased by two (Number of Nodes(Maximum Degree) = 20(6), 30(8), 40(10), 50(12), 60(14), 70(16), 80(18), 90(20), 100(22)). In the second simulation environment, value of maximum degree is not fixed according to the number of nodes. In case of both simulation environments, 30 simulations are performed for each steps and maximum, minimum and average values are computed for each step and used in performance evaluation and comparison.

The factors involved in the performance evaluation criteria are schedule length, latency and duty-cycle. The schedule length calculates the total number of timeslots required to assign a timeslot to each link. The latency shows the average number of timeslots after which a sender node can send a packet to same receiver node. Duty cycle means how many nodes are activated after completed schedule length.

A. SCHEDULING LENGTH

Fig. 11 shows the schedule length comparison of the proposed DDLS scheme against the DCLS scheme using first simulation environment, where maximum degree value is fixed for each step. Intuitively the schedule length increases with the increase in number of nodes as can be seen in the **Fig. 11**. In case of 20 nodes and maximum degree 6, the DCLS scheme requires maximum 62 timeslots; on average it requires 45 timeslots and minimum 10 timeslots. In contrast the proposed DDLS scheme requires maximum 18 timeslots; on average it requires 14 timeslots and minimum it requires 10 timeslots.

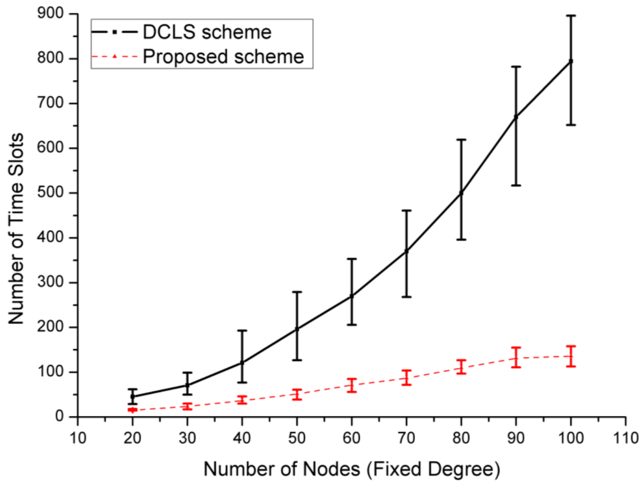


FIGURE 11. Scheduling length with increasing nodes and fixed maximum degree.

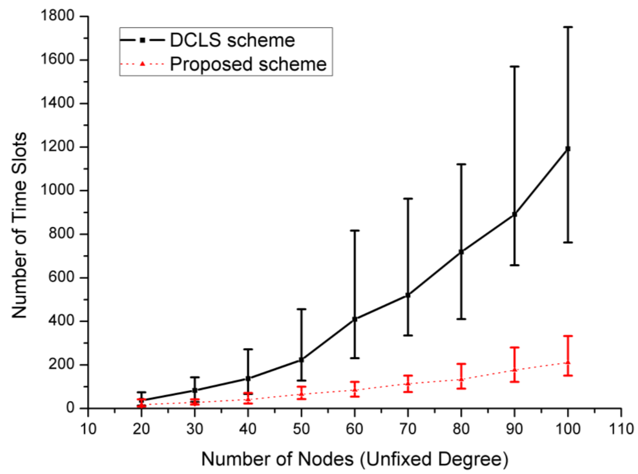


FIGURE 12. Scheduling length with increasing nodes and unfixed maximum degree.

In the **Fig. 11** the performance of the proposed DDLS scheme improves as the number of nodes increases. This is because the proposed DDLS scheme assigns the timeslots to link based on the CAMILS, which assign a same timeslot to as many links as possible with avoiding collision and interference. Overall the proposed DDLS scheme reduces the schedule length in comparison to DCLS scheme by 71% in case of maximum number of timeslots required and approximately 66% in case of average and minimum number of timeslots required. This considerable decrease in schedule length by the proposed DDLS scheme also improves the overall network delay.

Fig. 12 shows the result of schedule length using second simulation environment, where there is no limitation on the maximum degree value. The results in **Fig. 12** shows similar

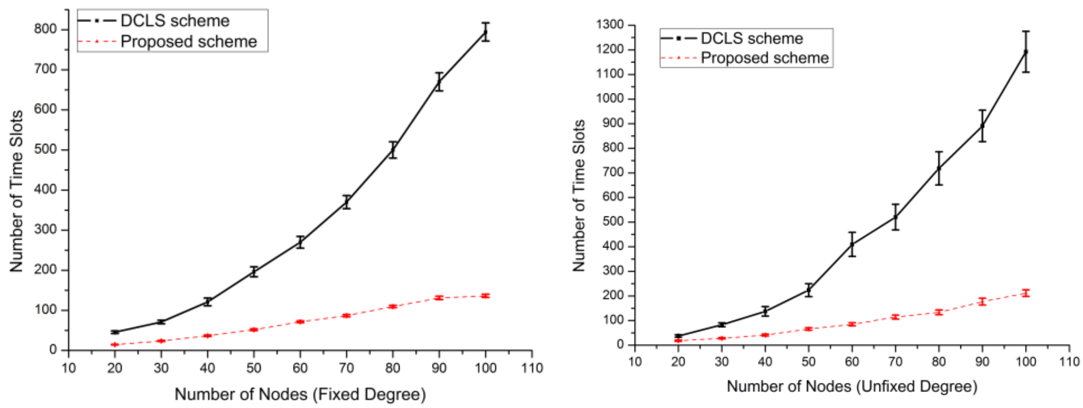
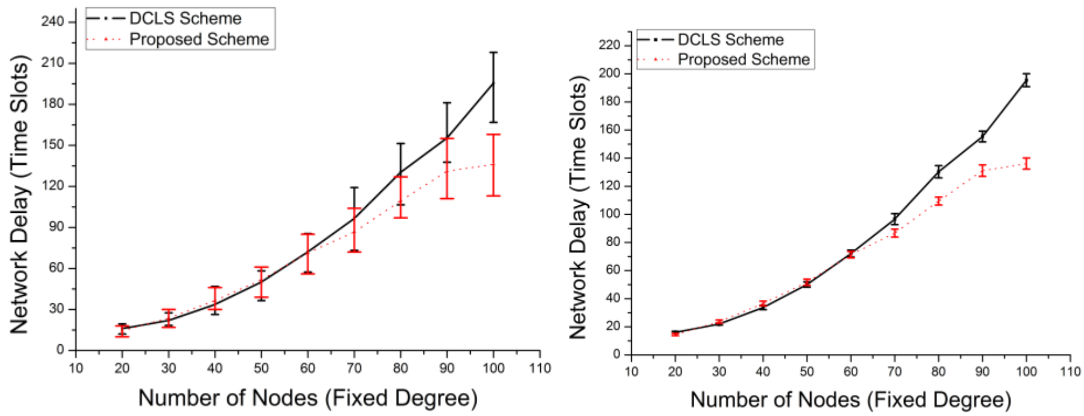


FIGURE 13. **Scheduling length with confidence interval 95%.**



(a) (b)

FIGURE 14. **Network delay with fixed degree. (a) The average delay. (b) The average delay with confidence interval 95%.**

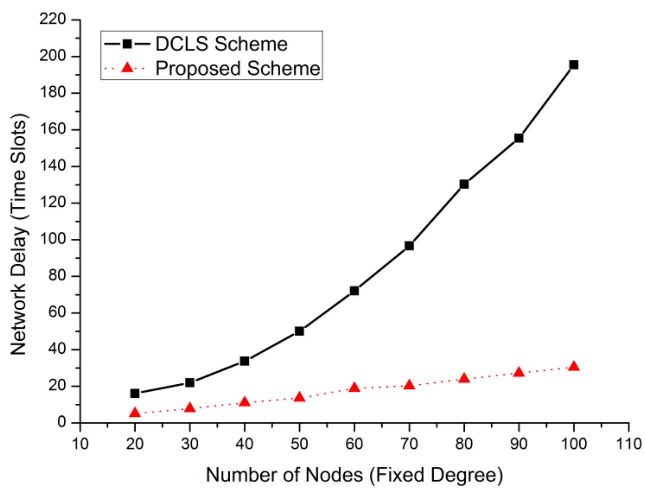


FIGURE 15. **The average delay in case of fixed degree with time normalization.**

pattern with **Fig. 11**. As there is no limitation on the maximum degree of a node therefore links of a node increases according to the density of the network, and hence require more timeslots to complete the schedule. For this reason single channel sensor node's performance is low in case of limitless maximum degree value comparing to fixed maximum degree value. However proposed DDLS scheme shows small gap in performance with and without limitation on maximum degree value. In case of DCLS scheme, if degree value of a node increases the total number of schedule length also increase rapidly.

The result in **Fig. 12** presents that when number of nodes are 20, proposed scheme's maximum value is 43%, average value is 52%, minimum value is 31% improved. When the nodes are 100, maximum value is 81%, average value is 82% and minimum value is 80% improved. Thus, with the increase in nodes, the performance gap also increases between two schemes. The proposed DDLS scheme is not relatively influenced by network environment comparing to DCLS scheme, as showed by maximum, average, minimum values result. Proposed scheme performance is stable and link scheduling is faster than DCLS scheme.

Fig. 13 shows the schedule length results in two simulations environments (with and without limit on maximum degree value respectively) with confidence interval of 95%. In case of first result in **Fig. 13** where maximum degree is limited, it is clear that gap between maximum and minimum

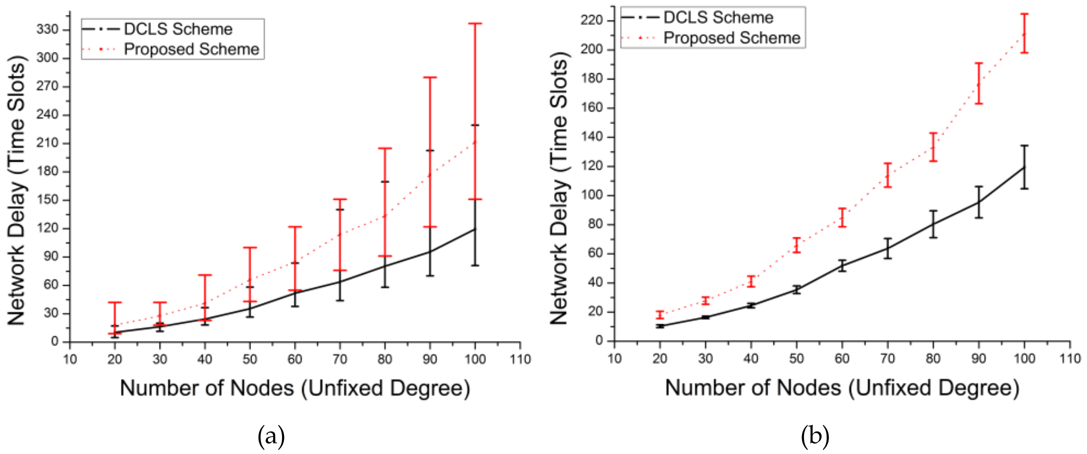


FIGURE 16. Network delay with unfixed degree. (a) The average delay. (b) The average delay a confidence interval 95%.

value is smaller than **Fig. 11's** result. The DCLS scheme depends on random factors in its algorithm which cause an unstable performance. But proposed scheme with confidence interval 95% shows total of schedule length similar to average value. In a case of second result in **Fig. 13** where there is no limit on the value of maximum degree, proposed scheme's schedule length doesn't show much variation from the average value. However DCLS scheme results show big gap between maximum and minimum values of schedule length with confidence interval 95%, which is much more than the case where maximum degree has limited value. Limited maximum degree value means that sensor nodes are uniformly dispersed. In contrast maximum degree value with no limit means that some specific region has more nodes than the other. In this case also proposed scheme's performance shows similar results as of Fig. 12 or first result in Fig. 13. The proposed DDLS scheme shows much more stable performance comparing to the DCLS scheme, which makes the DDLS scheme much more reliable and efficient in situations where sensor deployment is dynamic.

In this section comparison of network latency between DCLS and DDLS schemes is being drawn. Network delay (latency) represents the delay between two packets sent from a same sender node to same receiver node. Network delay (latency) for each node in the network is calculated to get the average latency of the whole network. The simulations are done for the two environments explained in earlier. **Fig. 14** shows the result of network latency in case of first simulation environment where value of maximum degree is limited. **Fig. 14(a)** presents the maximum, minimum and average values of network latency, and **Fig. 14(b)** shows the values of **Fig. 14(a)** with 95% confidence interval. When maximum degree is limited the proposed scheme shows similar results as DCLS scheme until nodes are 60. With more than 60 nodes proposed scheme starts to show lower latency comparing to DCLS scheme. **Fig. 14(b)** shows that proposed DDLS and DCLS scheme's maximum and minimum values are similar to average value with little variation. As established in

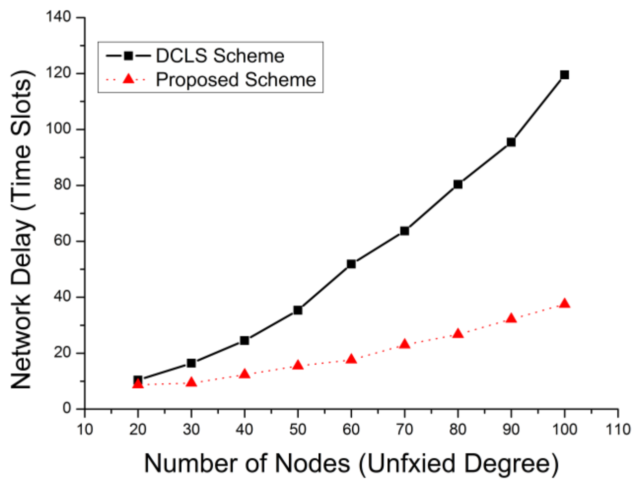


FIGURE 17. The average delay in case of unfixed degree with time normalization.

section V-A that schedule length in DCLS is longer than proposed scheme, however that result is not reflected in case of average network latency. This is because DCLS scheme assigns multiple timeslots to a link, therefore latency for some links is very less comparing to other links and overall average latency is also decreased. In case of proposed DDLS scheme each link is assigned one timeslot, therefore latency for each node is similar.

In order to further explain the network delay difference between DDLS and DCLS schemes with relation to schedule length, we make both schemes to have a same schedule length. **Fig. 15** shows the performance of two schemes when the schedule length is similar (time normalization), and vast difference is clear. The latency of proposed DDLS scheme is 88%$\sim$95% less than DCLS scheme. DCLS scheme reduces network latency by assigning multiple timeslots to a link. However, DCLS scheme assigns only one timeslot to the nodes which are remaining at the end. Thus, the latency increases as total schedule length.

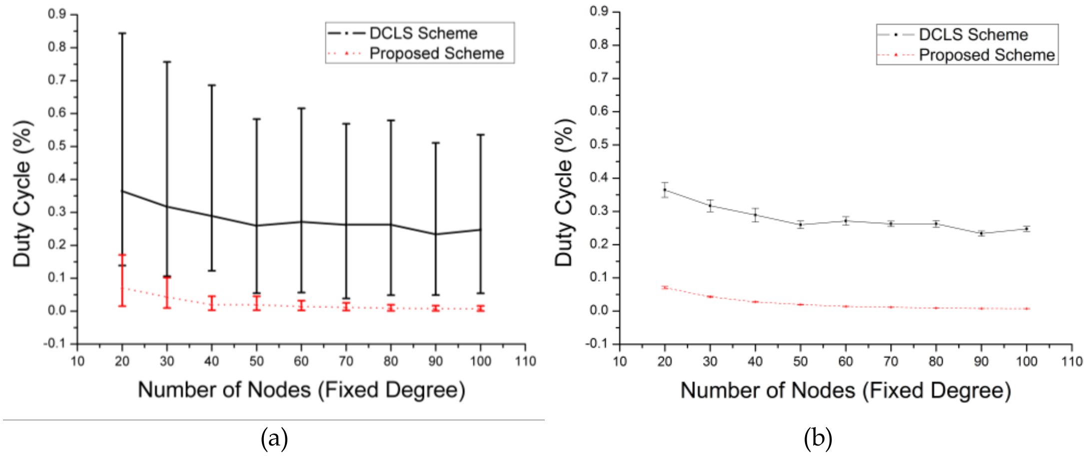


FIGURE 18. Duty Cycle comparison with the fixed degree value. (a) Duty cycle maximum-average-minimum. (b) Duty cycle with confidence interval 95%.

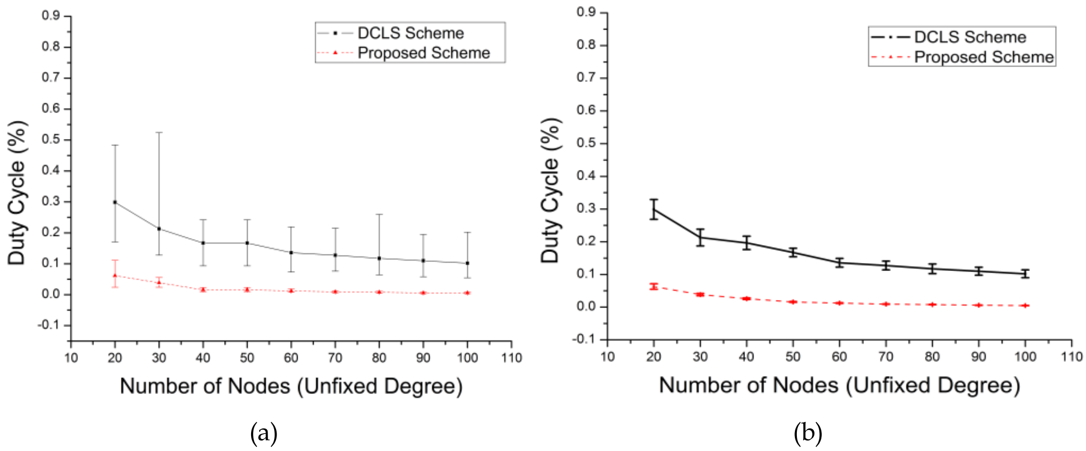


FIGURE 19. Duty Cycle comparison with the unfixed degree value. (a) Duty cycle maximum-average-minimum. (b) Duty cycle with confidence interval 95%.

Fig. 16 shows latency results from the simulation environment where there is no limit on the value of maximum degree. **Fig. 16 (a)** shows maximum, minimum, average values of network latency with unfixed maximum degree, and **Fig. 16(b)** shows the values of **Fig. 16(a)** with 95% confidence interval. These results show that the proposed scheme's latency is more than DCLS scheme and also in case of 95% confidence interval. The reason for this is because DDLS scheme assigns one timeslot to a link, but DCLS scheme assigns multiple timeslots to a link. In the experiment with fixed maximum degree, where each step increases 10 nodes and 2 in maximum degree value. The maximum degree is 14 when nodes are 60. Whereas in case of unfixed maximum degree the value of maximum degree is 16.3 for same number of nodes. This shows that when nodes are increasing, the fixed degree scenario does not truly represent the average maximum degree and is less. The proposed scheme assigns timeslots to unassigned links maximally. However, as Fig. 16 shows in case of unfixed maximum degree where a node has more neighbor nodes than fixed degree case, the proposed scheme results degrade.

Similar to **Fig. 15** if the schedule length of DDLS and DCLS scheme is same, which means two schemes perform their own algorithms in same time, then proposed DDLS scheme performance is far better, as shown in **Fig**. **17**. The proposed scheme's latency less than DCLS scheme about 68%∼84%. The performance improvement difference is less comparing to **Fig. 15**. If density of nodes is reduced than schedule length increase comparing to fixed maximum degree case.

B. DUTY CYCLE

Major portion of sensor node energy is spent on transmission. In this section, activation of nodes is calculated until the algorithm reaches completion. If the active nodes increase, the transmission of nodes will also increase which decrease the energy efficiency of the network. As a previous results show, the proposed DDLS scheme completes the link scheduling with smaller schedule length. **Fig. 18** shows the ration of active sensor nodes for performing algorithms between DDLS and DCLS schemes.

Fig. 18 (a) shows comparison result of maximum, average, and minimum value of the duty cycle with the fixed degree and **Fig. 18(b)** shows the 95% confidence interval of **Fig. 18(a)** results. Proposed scheme reduce duty cycle as maximum value 75%∼96%, average value 81%∼97%, and minimum value 89%∼95%. Thus, the active ratio of nodes is much lower than DCLS scheme, which improves the energy efficiency and life time of the WSN. The gap of proposed scheme's maximum and minimum values with confidence interval 95% is similar to the average value which shows the active ratio of nodes is stable.

The duty cycle comparison of the DDLS and DCLS schemes presented in the **Fig. 19** is consistent with the results of **Fig. 18**. It can be safely concluded from **Fig. 18** and **19** that degree value has no effect on the duty cycle in case of DDLS scheme, which shows that the proposed DDLS is stable in terms of dynamic deployment of the sensors. However the duty cycle values for DCLS scheme gets little better in the case of the unfixed degree.

VI. CONCLUSION

This paper presents a novel distributed TDMA based link scheduling scheme DDLS, which uses the degree information of the neighbors to calculate the sun of degree information and utilizes it to assign the timeslots to the links. Functioning of the proposed DDLS scheme is based on the CAMILS, through which primary and secondary collisions are avoided while assigning the timeslot to the link. Performance comparison of the proposed DDLS scheme has been made against the recently presented DCLS scheme, which clearly exhibits that the proposed DDLS scheme achieves substantial improvement in terms of schedule length and stability under the same network environment. In our current proposed DDLS scheme, only one timeslot is assigned per link regardless of the transmission requirement of the node. In the future work we extend the proposed DDLS scheme to support multiple timeslots for a link based on the amount of data a node has to transmit. This will require us to incorporate priority of node based on the data to transmit along with the current priority mechanism based on sum of degree of the node.

REFERENCES

[1] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, vol. 3. Jun. 2002, pp. 1567–1576.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[3] B. S. Krishnan, M. Ramaswamy, and A. Nachiappan, "Optimizing energy delay metric for performance enhancement of wireless sensor networks," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 5, pp. 1289–1297, 2010.

[4] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the efficiency of preamble sampling protocols for wireless sensor networks," in *Proc. IEEE MCWC*, vol. 1. Sep. 2006, pp. 117–122.

[5] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. ACM SenSys*, vol. 4. pp. 307–320, 2006.

[6] L. Shi and A. O. Fapojuwo, "TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 927–940, Jul. 2010.

[7] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2159–2167.

[8] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. ACM SenSys*, 2003, pp. 181–192.

[9] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *J. Wireless Netw.*, vol. 16, no. 4, pp. 985–997, 2010.

[10] Y. Wang and I. Henning, "A deterministic distributed TDMA scheduling algorithm for wireless sensor networks," in *Proc. IEEE WiCom*, Sep. 2007, pp. 2759–2762.

[11] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in sensor networks: Distributed edge coloring revisited," in *Proc. IEEE INFOCOM*, vol. 4. Aug. 2005, pp. 2492–2501.

[12] W. Wang, Y. Wang, X. Y. Li, W. Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw. ACM MobiCom*, 2006, pp. 262–273.

[13] C. Wang and K.-F. Ssu, "A distributed collision-free low-latency link scheduling scheme in wireless sensor networks," in *Proc. IEEE WCNC*, Apr. 2010, pp. 1–6.

[14] C. L. Barrett, G. Istrate, V. S. A. Kumar, M. V. Marathe, S. Thite, and S. Thulasidasan, "Strong edge coloring for channel assignment in wireless radio networks," in *Proc. IEEE PerCom Workshops*, Mar. 2006, pp. 105–110.

[15] M. Mahdian, "The Strong Chromatic Index of Graphs," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Ontario, Canada, 2000.

[16] J. Harris, J. L. Hirst, and M. Mossinghoff, *Combinatorics and Graph Theory*. Berlin, Germany: Springer-Verlag, 2008.

[17] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop TDMA wireless networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 870–883, Jun. 2009.

[18] C. Godsil and G. F. Royle, *Algebraic Graph Theory*. Cambridge, UK: Springer, 2001.

[19] K. Koushik, S. Sarkar, A. Ghavami, and X. Luo, "Delay guarantees for throughput-optimal wireless link scheduling," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2906–2911, Nov. 2012.

[20] N. A. Pantazis, D. J. Vergados, D. D. Vergados, and C. Douligeris, "Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling," *J. Ad Hoc Netw.*, vol. 7, no. 2, pp. 322–343, 2009.

[21] V. Cionca, T. Newe, and V. Dadârlat, "TDMA protocol requirements for wireless sensor networks," in *Proc. IEEE SENSORCOMM*, Aug. 2008, pp. 30–35.

[22] I. Slama, B. Jouaber, and D. Zeghlache, "A free collision and distributed slot assignment algorithm for wireless sensor networks," in *Proc. IEEE GLOBECOM*, Dec. 2008, pp. 1–6.

[23] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: A survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, Apr. 2006.

[24] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE J. Netw.*, vol. 20, no. 3, pp. 20–25, May/Jun. 2006.

[25] S. S. Kulkarni, "TDMA services for sensor networks," in *Proc. IEEE 24th Int Conf. Distrib. Comput. Syst. Workshops*, Mar. 2004, pp. 604–609.

[26] A. Bachir and M. Dohler, "Mac essentials for wireless sensor networks," *IEEE Commun. Surv. Tuts.*, vol. 12, no. 2, pp. 222–248, 2nd Quart.,2010.

[27] S. Myoung, D. Kim, H. Choo, and M. Kim, "Degree-based link scheduling for collision avoidance in distributed wireless sensor networks," in *Proc. 8th Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, 2014, pp. 116:1–116:6.

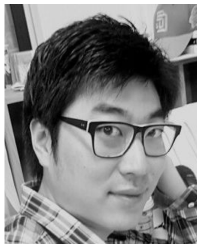
[28] S.-G. Jung, B. Kang, S. Yeoum, and H. Choo, "Trail-using ant behavior based energy-efficient routing protocol in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2016, pp. 1–9, 2016.

[29] B. Kang, N. Kwon, and H. Choo, "Developing route optimization-based PMIPv6 testbed for reliable packet transmission," *IEEE Access*, vol. 4, pp. 1039–1049, Mar. 2016.

[30] B. Kang and H. Choo, "A cluster-based decentralized job dispatching for the large-scale cloud," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–8, Dec. 2016.

[31] J. Dams, M. Hoefer, and T. Kesselheim, "Scheduling in wireless networks with rayleigh-fading interference," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1503–1514, Jul. 2015.
[32] F. Al-dhelaan, P. J. Wan, and H. Yuan, "A new paradigm for shortest link scheduling in wireless networks: Theory and applications," in *Proc. Int. Conf. Wireless Algorithms, Syst. Appl.*, 2016, pp. 24–36.
[33] H. Deng, J. Yu, G. Li, Y. Dongxiao, and B. Huang, "Heuristic algorithms for one-slot link scheduling in wireless sensor networks under SINR," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 3, p. 806520, 2015.

BYUNGSEOK KANG received the B.S. degree in computer engineering from Sejong University, South Korea, in 2006, the M.S. degree in electrical and electronics engineering from Korea University, South Korea, in 2008, and the Ph.D. degree in electronics and computer science from the University of Southampton, U.K., in 2015. He is currently a Research Professor with Sungkyunkwan University in 2015. His research interests include wired/wireless networking, sensor networking, mobile computing, network security protocols, and simulations/numerical analysis.



SUNGHO MYUNG received the M.S. degree in information and communication engineering from Sungkyunkwan University, South Korea, in 2014. He is currently a Researcher with LG Electronics, 917-1, Jeongjo-ro, Jangan-gu, Suwon-si, Gyeonggi-do, Republic of Korea, 16270. His research interests include wireless ad-hoc networks, wireless senor networks (WSNs), and network performance evaluation.

HYUNSEUNG CHOO received the B.S. degree in mathematics from Sungkyunkwan University, South Korea, in 1988, the M.S. degree in computer science from the University of Texas at Dallas, USA, in 1990, and the Ph.D. degree in computer science from the University of Texas at Arlington, USA, in 1996. From 1997 to 1998, he was a Patent Examiner with the Korean Industrial Property Office. Since 1998, he has been with the College of Information and Communication Engineering, Sungkyunkwan University, and has been an Associate Professor and the Director of the Convergence Research Institute. Since 2005, he has been the Director of the Intelligent HCI Convergence Research Center (eight year research program) supported by the Ministry of Knowledge Economy, South Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment. He has authored over 200 papers in international journals and refereed conferences. His research interests include wired/wireless/optical embedded networking, mobile computing, and grid computing. He is member of the ACM. He is the Vice President of the Korean Society for Internet Information (KSII). He has been the Editor-in-Chief of the Journal of KSII for three years and journal editors of the *J*ournal of Supercomputing, the *I*nternational Journal of Mobile Communication, and the *S*pringer-Verlag Transactions on Computational Science Journal, and has been an Editor of the *K*SII Transactions on Internet and Information Systems since 2006.