



FAJIT: a fuzzy-based data aggregation technique for energy efficiency in wireless sensor network

Shashi Bhushan¹ · Manoj Kumar² · Pramod Kumar³ · Thompson Stephan⁴ · Achyut Shankar⁵ · Peide Liu⁶

Received: 25 July 2020 / Accepted: 10 December 2020 / Published online: 8 January 2021
© The Author(s) 2021

Abstract

Wireless sensor network (WSN) is used to sense the environment, collect the data, and further transmit it to the base station (BS) for analysis. A synchronized tree-based approach is an efficient approach to aggregate data from various sensor nodes in a WSN environment. However, achieving energy efficiency in such a tree formation is challenging. In this research work, an algorithm named fuzzy attribute-based joint integrated scheduling and tree formation (FAJIT) technique for tree formation and parent node selection using fuzzy logic in a heterogeneous network is proposed. FAJIT mainly focuses on addressing the parent node selection problem in the heterogeneous network for aggregating different types of data packets to improve energy efficiency. The selection of parent nodes is performed based on the candidate nodes with the minimum number of dynamic neighbors. Fuzzy logic is applied in the case of an equal number of dynamic neighbors. In the proposed technique, fuzzy logic is first applied to WSN, and then min–max normalization is used to retrieve normalized weights (membership values) for the given edges of the graph. This membership value is used to denote the degree to which an element belongs to a set. Therefore, the node with the minimum sum of all weights is considered as the parent node. The result of FAJIT is compared with the distributed algorithm for Integrated tree Construction and data Aggregation (DICA) on various parameters: average schedule length, energy consumption data interval, the total number of transmission slots, control overhead, and energy consumption in the control phase. The results demonstrate that the proposed algorithm is better in terms of energy efficiency.

Keywords Wireless sensor network · Fuzzy logic · Data aggregation · Scheduling · Tree formation

Introduction

Wireless sensor networks can be homogeneous or heterogeneous. When an application requires measurement of a single physical quantity, then the network is considered as homogeneous, otherwise, it is a heterogeneous network. Scheduling and tree formation in a heterogeneous network is different than in the homogeneous network. Many of the existing

works in WSN and its variants [36] have focused on aggregation [10,15], energy consumption [8,38], mobility [40], data processing methods [41], and scheduling [42] techniques.

In tree-based network [37], non-leaf nodes receive one or more packets from children nodes. If network is homogeneous, every node can aggregate all incoming packets with its own packets. As a result, only one packet goes out of the

✉ Thompson Stephan
thompsoncse@gmail.com

Shashi Bhushan
tyagi_shashi@yahoo.com

Manoj Kumar
wss.manojkumar@gmail.com

Pramod Kumar
drpramoderp@live.com

Achyut Shankar
ashankar2711@gmail.com

Peide Liu
peidi.liu@gmail.com

¹ School of Computer Science and Engineering, University of Petroleum and Energy Studies, Dehradun, India

² School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

³ Krishna Engineering College, Ghaziabad, India

⁴ Department of Computer Science and Engineering, Faculty of Engineering and Technology, M. S. Ramaiah University of Applied Sciences, Bangalore 560054, Karnataka, India

⁵ Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India

⁶ Shandong University of Finance and Economics, Jinan, China

node. Homogeneous network has all temperature sensors. Every node sends and receives packets of type temperature. At a given node, all temperature packets may be aggregated. Aggregation function can be sum, average, median, etc., thus only one packet goes out. As a result, every node has to select one parent and one time slot. Data aggregation is how data coming from different sensors are combined and provides useful aggregated information. Some of the existing works [2,3,16] have already improved the data aggregation/energy efficiency of WSN without using any AI or fuzzy-based techniques [34,35]. However, the work presented in this paper uses a fuzzy logic technique which is very efficient in real-time systems and can deal with dynamic situations and model the inherently imprecisely defined conditions.

In a heterogeneous network, different types of nodes are present in the network [28]. This type of network has temperature and pressure sensors. There is a possibility that the children of a given node have a different type. Thus, it might receive different types of packets. On improvement of the aggregation technique, the average number of packets coming out per node decreases. This results in a reduction in the required number of time slots and energy consumption. If aggregation is improved in the heterogeneous network, the transmission of packets will become less as compared to DICA and thus the count of total transmission slots is reduced. Finally, schedule length, energy consumption are reduced. At every node, energy is consumed owing to the following reasons: (1) control packets exchanged with neighbors during slot and parent selection. (2) Transmission of data packets.

Suitable parent selection reduces energy consumption during both the control phase and the data transmission phase. To minimize energy efficiency [32], the node has to select less number of slots for which fewer control messages are needed. As fewer data packets are to be sent, energy consumption during the data phase is reduced. As explained in DICA [6], slot and parent selection (i.e. tree formation) should take place jointly. Initially, the tree is formed followed by the slot assignment. The tree structure controls the performance of the scheduling algorithm [22]. The scheduling algorithm is not dependent on the tree structure as both scheduling and tree formation are performed together. In other words, every node must check if there is any suitable parent in the lowest available time-slot. Instead of selecting the parent first and then finding time-slot, it is better to select slot and parent together such that the node can transmit in the lowest possible slot. This approach reduces the schedule length of the tree. Therefore, the joint approach is applicable to heterogeneous networks.

When the network is heterogeneous, different types of nodes are present in the system. As mentioned earlier, bottom-up scheduling and parent selection are desirable to maintain aggregation freshness in aggregated converge cast.

When a node attempts to decide its slot and parent, it has the following pieces of information: (1) number of incoming packets, and (2) type of each incoming packet. Moreover, the node also knows the type of packet generated by itself. The node can identify the following: (1) number of outgoing packets, and (2) type of each outgoing packet. For each outgoing packet, the node may select a different parent such that the packet would be aggregated as soon as possible. Parent selection based on the count of unscheduled neighbors is not a suitable approach for heterogeneous networks.

The main contribution of this research work are listed as follows:

- Proposes and implements the fuzzy attribute-based joint integrated scheduling and tree formation (FAJIT) algorithm for single-sink heterogeneous networks to improve energy efficiency in WSN.
- Selects a different parent for every outgoing packet.
- Performs parent selection based on the type of packet.

The problem of scheduling and parent selection in heterogeneous networks is formally defined in the following sections. The paper is divided into five sections. The next section discusses the literature review followed by which the idea of the proposed algorithm is discussed. The subsequent section discusses the results of the proposed approach, and the final section concludes the paper.

Related work

This research work focuses on distributed scheduling and tree formation algorithms in single-sink and multi-sink networks [9]. In the first subsection, distributed scheduling algorithms for single sink networks are presented. Then the mechanisms related to fault tolerance are discussed. The discussed categories are (1) algorithms addressing aggregated converge cast, (2) algorithms addressing raw converge cast, (3) algorithms that can be adapted for use with any of the two types of converge cast, i.e. general algorithms.

Algorithms addressing aggregated converge cast assign single transmission slot to every node. Slot assignment is preferred to be bottom to top, i.e. from leaf to root [33]. By considering aggregated converge cast, all children packets are aggregated with parents and a single packet is formed which will be transmitted. If the time slot assigned to the parent is lower than the children, the parent can forward the aggregated packet only in the next TDMA cycle. When the parent is assigned a higher time slot, the aggregated packet can be forwarded in the same cycle. Therefore, packet latency can be controlled by the bottom-top slot assignment. Algorithms categorized under the general category do not address any specific type of converge cast and most of them are

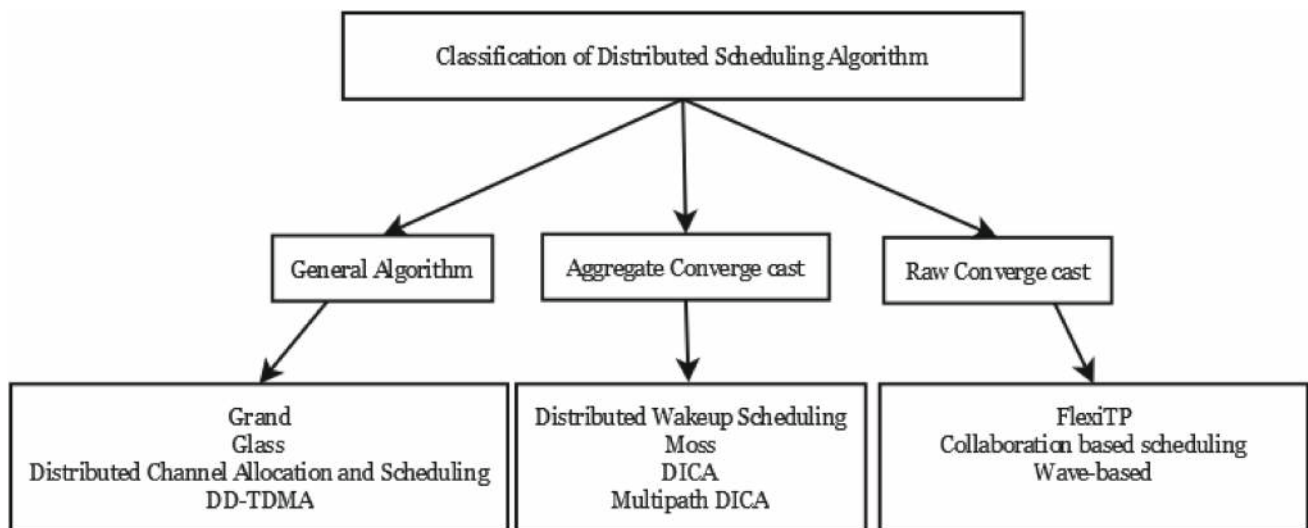


Fig. 1 Flow of distributed scheduling algorithm

not designed for tree-based networks. But they are targeted towards other issues such as reducing control overhead of slots selection, and use of multiple channels for better slot reuse. The scheduling should take place in a bottom-up manner in an aggregated converge cast. In raw converge cast, scheduling should be done in a top-down fashion. The methods categorized as general methods are not tuned to any specific converge cast. The classification of different scheduling algorithms is shown in Fig. 1.

DICA [6] is more appropriate for the following reasons: (1) in DICA, the selection of parent is done by the node in such a way that it can be transmitted in the smallest possible time slot. The parent may be at the same level as a given node, one-hop near to sink, or maybe one-hop far from the sink. In the other two approaches, the parent must be one-hop near to sink than the given node. As DICA is focused on selecting any neighbor as a parent which can receive in the smallest selected slot, it is likely to result in the smallest schedule length. The distributed scheduling algorithm proposed in [17] uses a sequential approach and it works in a top-down manner. The top-down approach is not suitable for heterogeneous networks. In heterogeneous networks, all incoming packets need not be aggregated with the given node's packet and hence, the multiple packets can come out from the given node. To transmit the packet, the total number of incoming packets along with their types is calculated first, so that in transmission, the details of the outgoing packets and their types can be identified. This is possible only if nodes are scheduled from leaf to sink, i.e. bottom to top. Therefore, in heterogeneous networks, bottom-up scheduling is more appropriate compared to top-down scheduling. Scheduling and parent selection should be done separately in heterogeneous networks.

This research work focuses on designing a joint scheduling and parent selection algorithm for heterogeneous networks. The objective of the proposed work is to maximize the aggregation. If schedules are not balanced, nodes in one tree would wait for a long time to get their turn to transmit, and as a result packet latency increases. On the other hand, a tree with a small schedule length produces low latency. If schedule lengths are balanced, nodes of both the trees would suffer equal packet latency. The overall schedule length (S^H) of the network would be $\max(S_1^H, S_2^H)$. Therefore, balancing the schedule lengths of individual trees reduces the overall schedule length. The other reason for the difference in the schedule lengths of the trees is the different levels of heterogeneity present in different regions of the network. For example, if there are two regions, one region is having two types of nodes and the other region is having six types of nodes. The region with two types of nodes is likely to result in better aggregation compared to the other region. As a result, the tree passing through the region with two types of nodes has a smaller schedule length than the other tree passing through a region having six types of nodes.

Most of the papers in the area of load balancing try to reduce the funneling effect [12] or distribute workload across one-hop nodes of the sender, and provide dynamic load balancing [14]. In addition, scheduling is not implemented by most of the researchers. Sia et al. [31] aims is to balance the load across the sub-trees. When nodes are not distributed uniformly, the load may be balanced across sub-trees present in the dense region. These sub-trees may be part of a single tree. When tree present in the dense region is scheduled, its schedule length is likely to be more than that of the tree formed from nodes present in the sparse region. This work does not focus on schedule length balancing of trees. In addition, it is a

centralized algorithm and does not attempt scheduling. Yu et al. [43] proposed an algorithm to divide the entire region into Voronoi sub-regions considering the sink nodes. In the case of non-uniform distribution of nodes, some Voronoi regions results in more number of nodes, and others with less number of nodes. Therefore, trees and corresponding schedules are likely to be unbalanced.

A distributed algorithm in multi-sink sensor networks for schedule length balancing of trees is proposed. The reason for unbalanced schedule length can either due to uneven distribution of nodes or difference in heterogeneity between different regions of the network. From the literature, two gaps are identified and a solution is provided in the proposed approach:

1. Present aggregation converge cast scheduling algorithms assumes the presence of the homogeneous network. Existing algorithms should be modified to take heterogeneity into account with the objective of maximizing aggregation.
2. Various algorithms are already present but most of them try to eliminate the funneling effect. There is no algorithm present in the literature addressing fuzzy logic to balance schedule length from different sinks in a tree.

Proposed algorithm

The authors in [37] implemented a homogeneous network for joint distributed scheduling and tree formation. It is modified to work with heterogeneous networks. The proposed fuzzy attribute-based joint integrated scheduling and tree formation (FAJIT) is discussed in this section. When the network is heterogeneous [39], different types of nodes are present in the network. As discussed earlier, bottom-up scheduling and parent selection are desirable to maintain aggregation in aggregated converge cast. When a node attempts to decide its slot and parent, the following activity needs to be performed: (1) determining the number and type of incoming packets, (2) labeling all the nodes, (3) adding weights to all the edges. Here, the weight is decided considering the distance between the nodes. Eq. 1 represents the aggregation factor (η_i) for node i and the average aggregation factor is represented in Eq. 2. For each outgoing packet, a node can select a different parent such that packet would be aggregated as early as possible.

$$\eta_b = \frac{R_i - F_i}{R_i}, \tag{1}$$

$$\eta = \sum_{i=1}^n \frac{\eta_i}{n}, n = 1, 2, 3, \dots, n. \tag{2}$$

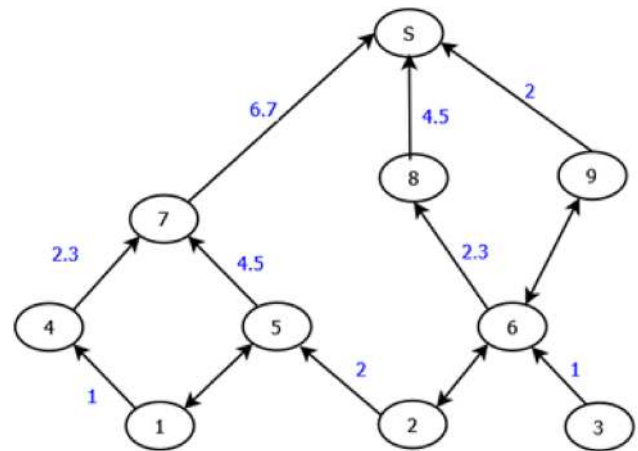


Fig. 2 Illustration of scheduling and tree formation using DICA_EXTENSION

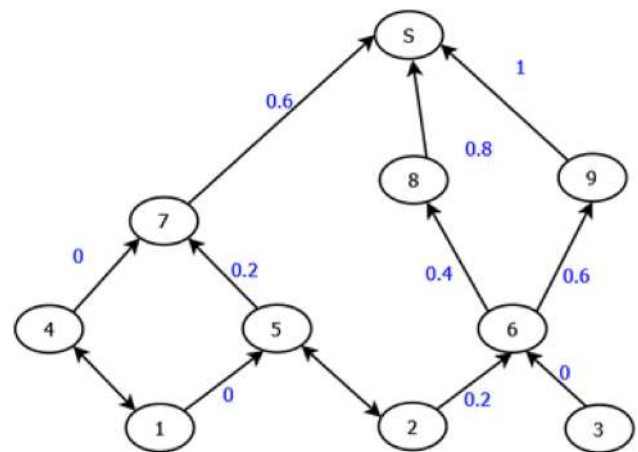


Fig. 3 Illustration of scheduling and tree formation using FAJIT

Figures 2 and 3 comprise of a heterogeneous network [23,25,29]. Two types of sensors, i.e. temperature and pressure are present. Figure 2 illustrates scheduling and parent selection without considering node heterogeneity. DICA is used for slot/parent selection. Figure 3 illustrates scheduling and parent selection as per FAJIT. We have applied min–max normalization to fuzzify the network and this is achieved using the following equation:

$$V' = \frac{V - \text{Min}(A)}{\text{Max}(A) - \text{Min}(A)} (\text{newMax}(A) - \text{newMin}(A)) + \text{newMin}(A). \tag{3}$$

In FAJIT, scheduling and tree formation is based on labeling and uses a bottom-up approach. First, calculate normalized weights to all the edges based on the min–max normalization. The following approach is designed for parent node selection.

Parent node selection

Find the candidate for the parent set by selecting the nodes having a direct link with the child node. Further from the desired parent set, find the number of dynamic neighbors of that candidate set [27]. To choose the parent node, count the number of dynamic neighbors, and the candidate set with the minimum number of dynamic neighbors will be selected as the parent node.

However, a scenario may arise where two candidate sets having the same number of dynamic neighbors, and in that case, the selection of the parent node becomes difficult. To overcome this, we first fuzzify the wireless sensor node [11] graph and then apply min–max normalization to estimate the normalized weight on the edges of the graph. Weights on the edges act as membership values. Membership value denotes the degree to which an element belongs to a set hence the node with the minimum sum of all the weights directly in contact with that node is declared as the parent node.

Each node should perform the following steps to find a parent for forwarding a packet of type t in a given time slot as per the proposed method:

1. Check if there is any neighbor of type t in the neighborhood. In that case, the packet should be sent to that neighbor. If no such node is found then execute step 2.
2. Check if there is any node in the neighborhood which is receiving packets of type t from other nodes. If any such node is found, it should be considered as the parent for that packet. If no such node is found then step 3 is executed.
3. Check if there is any node in the neighborhood which has one or more nodes of type t in its neighborhood. If any such node is found, it should be considered as the parent for a packet of type t . If no such node is found, then step 4 is executed.
4. Select the parent with the minimum number of unscheduled neighbors as a neighbor node.

The FAJIT algorithm is described in Algorithm 1. The notations used in FAJIT are shown in Table 1. Input to the algorithm is a message m . The output of the algorithm is a parent node and a slot to transmit that message.

The sequence of operations involved in FAJIT algorithm is elaborated as follows:

1. Graph formation
 - (a) Initialize the node graph with N_{Root} .
 - (b) Add and label the nodes.
 - (c) Add weights to all the edges.
 - (d) The weights will be decided by the distance between the two nodes.
2. Parent node selection

Algorithm 1: FAJIT

```

/* Initialize the node graph with  $N_{Root}$  */
1 if  $root = \emptyset$  then
2   | Create Root Node  $N_{Root}$ 
3 end
/* Adding Nodes to Tree */
4 while true do
5   | Using Function AddEdge (int Source, int Dest, int Weight)
   | /* Source & Dest are labels of nodes and
   |    Weight is the distance between nodes
   |    */
6 end
/* Nodes with direct link to the child node
*/
7 for  $i = 1$  to  $N$  do
8   | if  $n[i] > TS_u$  and  $n[i] \leq TM_u$  then
9     | |  $n[i] \in TS_u$ 
10    | end
11 end
/* Using Min Max Normalization */
12 for  $i = 1$  to  $N$  do
13   | while  $n[i] \in TS_u$  do
14     | | if  $Distance(Source, n[i]) \leq TR$  then
15       | | |  $V' = \frac{V-Min}{Max-Min} (newMax - newMin) + newMin$ 
16       | | end
17     | end
18 end
/* Selection of Parent Node */
19 for  $i = 1$  to  $N$  do
20   | for  $i = 1$  to  $TS_u$  do
21     | | if  $Distance(S, n_j) > Distance(S, n_j + 1)$  then
22       | | | ParentNode= $n_j$ 
23     | | else
24       | | | ParentNode= $n_j + 1$ 
25     | | end
26   | end
27 end

```

- (a) Find the candidates for the parent set i.e. nodes which has direct link to the child node.
- (b) From the desired parent set, find the number of dynamic neighbors of that candidate node.
- (c) The candidate node with minimum number of dynamic neighbors will be considered as the parent node.
- (d) If the number of dynamic neighbors are equal for both the candidate sets then follow the following steps:
 - (i) Fuzzify the given wireless sensor node graph.
 - (ii) Use Min-Max normalization to retrieve normalized weights for given edges of the graph.
 - (iii) These weights will now act as the membership value of the edges.
 - (iv) Since membership value denotes the degree to which an element belongs to a set, the node with the minimum sum of all the weights directly in contact with that node will be taken as the parent node.
3. Testing

Table 1 Notations used in FAJIT algorithm

Parameter	Meaning
P_u	Parent of node u
TS_u	Transmission slot of node u
TM_u	Longest transmission slot of children of node u
OH_u	Neighbors of node u overhearing in TS_u
RH_u	Neighbors of node u receiving in TS_u
TR_u	Neighbors of node u transmitting in TS_u
$candp_u [TS_u]$	Neighbors of node u which do not transmit, receive or overhear in slot TS_u
E^C	Energy consumption during control phase
E^D	Energy consumption during data phase
E^{eres}	Residual energy at end of control phase [30]
E^{init}	Initial energy of every node
E^{dres}	Residual energy at the end of data phase [30]
<i>REQUEST</i>	The message broadcast by a node to inform its choice of slot and parent to its candidate parents
<i>REPLY</i>	Response generated by candidate parents upon receiving <i>REQUEST</i> message
<i>SCHEDULE</i>	The message broadcast by a node to confirm its choice of slot and parent to its candidate parents
<i>FORBIDDEN</i>	Message broadcast by a candidate parent upon reception of <i>SCHEDULE</i> message

- Further, test the efficiency of the proposed algorithm with respect to parameters like energy consumption and control overhead on a simulated graph.
- Plot the results for better visualization and analysis in *NOT READY* state. It calls function *not ready ()*. l is the level for a given node. For scheduling, it has to wait for its neighbors in level $(l + 1)$. When all such nodes are scheduled, the given nodes switches to *READY* state. It calls function *ready ()*.

In *ready ()* function, the node selects the lowest possible transmission slot TS_u as one more than the highest transmission slot of children. TS_u will be incremented by node until TS_u is found such that no neighbors are receiving in TS_u , and the set of candidate parents is not empty. In the next step, the *parent selection ()* function is called to select a suitable parent. For illustration, Node A is forwarding a packet of type t . First, it checks if there is a node of type t in the candidate parent set. If such a node is present, it is selected as parent and function returns. Otherwise, the node receiving the maximum number of packets of type t is selected as parent and function returns. Node checks if there is any candidate parent which has any neighbor of type t , and if true, such a node is selected as a parent.

Performance evaluation, results and discussion

This section evaluates the performance of FAJIT through extensive simulations and present simulation setup and

parameters, performance metrics, and performance evaluation.

Simulation setup

A square area of 3000×3000 m is considered for node deployment. Nodes are deployed randomly. Grid of 20×20 points is formed to divide the area into grids. A fixed distance of 156 meters is selected between every two horizontal and vertical grid points. Nodes are probabilistically deployed at grid points. The value of the Probability P_d is 0.5, i.e., there is 50% chance that a node is present in a given grid point.

Four different scenarios are presented in this section to discuss the performance of FAJIT. In the case of the number of attributes being 1, all the nodes are of the same type. In the first scenario, the number of attributes present in the network is 2. In the second scenario, the number of attributes is 4. It means that four types of nodes are present in the network. For more number of attributes, the network becomes more and more heterogeneous.

Every node is implemented with randomly assigned attributes and the number of attributes is denoted as n_A . All attributes are of equal probability. That is, the probability P^{A_i} for node i is assigned attribute A_i is as per Eq. 4:

$$P^{A_i} = \frac{1}{\eta_i} \quad (4)$$

For example, in the second scenario, n_A is 4. Four attributes A_1, A_2, A_3, A_4 , and estimated probability P^{A_i} is 0.25, i.e., a node is assigned any one attribute with probability 0.25. As

Table 2 Simulation setup

Parameter	Value
Node deployment	Random
Number of nodes	300
Radio radius	30m
Transmission power consumption	0.660 w
Received power consumption	0.395 w
Sleep power consumption	0 w
Set simulation time	2500 sec

attribute denotes the type of node, i.e., A_1 is temperature, A_2 is pressure, A_3 is solar radiation, and A_4 is humidity. The performance of the proposed method is evaluated by varying heterogeneity levels of the networks. The parameters control overhead (CO), average energy consumption in the control phase (EC), and average energy consumption in the data phase (ED) are all estimated using the following equations: respectively.

$$C^O = \sum_{i=1}^n C_i^O, i = 1, 2, 3, \dots, n, \tag{5}$$

where C_i^O denotes the control overhead at node i .

$$E^C = \frac{\sum_{i=1}^n E_i^C}{n}, i = 1, 2, 3, \dots, n, \tag{6}$$

where $E_i^C = E_i^{\text{init}} - E_i^{\text{cres}}$.

$$E^D = \frac{\sum_{i=1}^n E_i^D}{n}, i = 1, 2, 3, \dots, n, \tag{7}$$

where $E_i^D = E_i^{\text{cres}} - E_i^{\text{dres}}$.

For every scenario, five different instances are generated randomly. Simulation results for any scenario are an average of results generated in different simulation runs for a particular scenario. In this paper, for the simulation scenario, four different scenarios have been taken with a varying number of attributes (n_a) as 2, 4, 6 and 8.

Performance metrics

The simulation setup is shown in Table 2. The transmission range of the nodes is set to 30 m and it is the same for all the sensor nodes. Every node generates one data packet every 10 s. The duration of the simulation is 2500 s. The initial 2000 s are allotted for the control phase. During the control phase, nodes perform slot and parent selection. The rest 500 seconds are used for the data phase. Nodes are generated and data packets are sent during the data phase through the tree

which is formed during the control phase. The following performance metrics are assessed through network simulation.

1. Schedule length (SH)
2. Average aggregation factor (η)
3. Control overhead (C^O)
4. Average energy consumption in the control phase (E^C)
5. Average energy consumption in the data phase (E^D)

Performance evaluation

According to Fig. 4, on comparing FAJIT technique with DICA and DICA_EXTENSION in terms of average aggregation factor, FAJIT scores better. Through fuzzification [7] and min–max normalization, network complexity becomes low as nodes with 0 membership values are ignored. With fewer nodes, fewer packets are generated which results in good aggregation percentage. When the network is homogeneous, perfect aggregation occurs in all the algorithms. Every node sends only one packet irrespective of how many packets are received [5]. Therefore, the estimated aggregation factor is 1 (100% aggregation). As the number of attributes increases, the aggregation factor decreases. It is observed that when two attributes are present, the difference between aggregation factors of FAJIT and DICA_EXTENSION is around 40%. But, as heterogeneity increases, the gap between aggregation factors of the two methods is reduced. When the number of attributes is 4, the difference is around 33.33%. In the end, the gap is as small as 10% when the number of attributes is increased to 16. When heterogeneity is high, it is difficult for FAJIT to find a parent where packets could be aggregated.

The packet gets aggregated after traveling more number of hops [4]. Therefore, at a high level of heterogeneity, the performance of FAJIT is approximately close to DICA_EXTENSION. Figure 4 depicts that the aggregation factor decrease with the increase in the number of attributes. On average, considering all the attributes, FAJIT maximizes the average aggregation factor by 36.77% and 18.22%, as compared to DICA and DICA_EXTENSION, respectively.

The results of schedule length with regard to the number of attributes are shown in Fig. 5. It is observed that a small schedule length is needed by FAJIT as compares to DICA and DICA_EXTENSION. Due to better aggregation, the number of packets passing through the tree during a TDMA cycle is reduced [13]. Hence to schedule the tree, the slot requirement is minimized. FAJIT results in smaller schedule length than DICA_EXTENSION. When the number of attributes is 8 and 10, on average, FAJIT gives a 4.84% smaller schedule length than DICA_EXTENSION and this increases with the increase in the number of attributes. FAJIT provides better results as the number of packets to be transmitted are less and less energy is required. On average, considering all the attributes, FAJIT minimizes the average

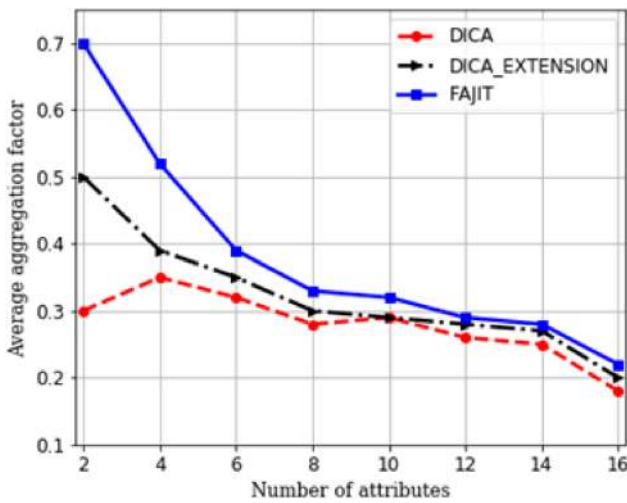


Fig. 4 Average aggregation factor versus number of Attributes

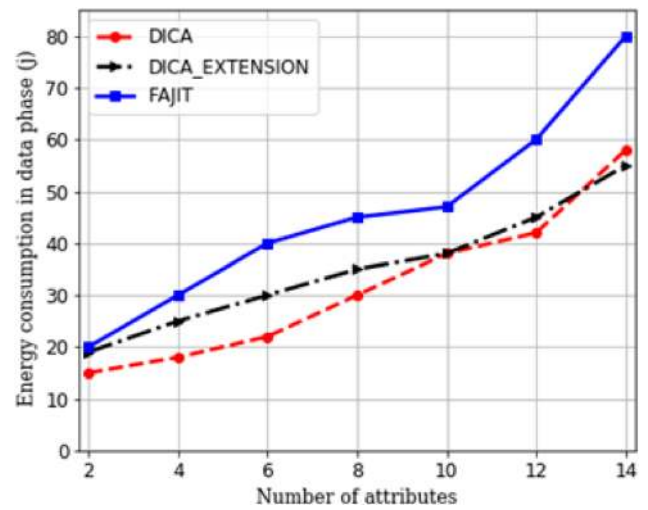


Fig. 6 Energy consumption in data phase versus number of attributes

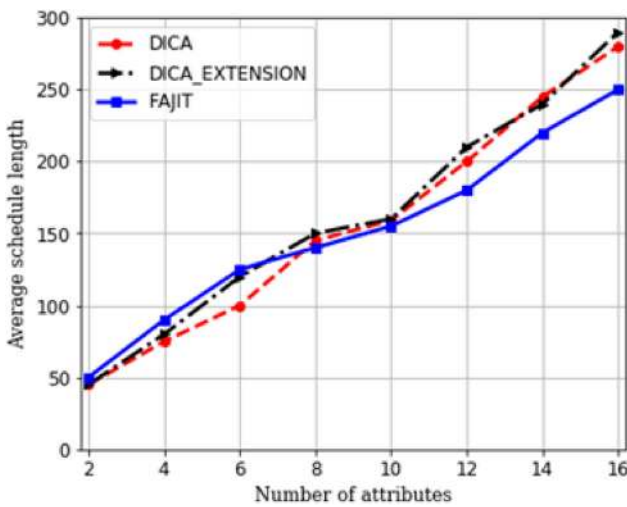


Fig. 5 Average schedule length versus number of attributes

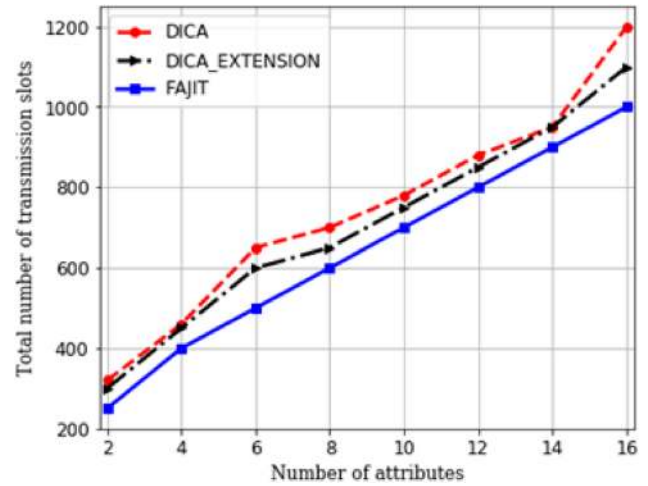


Fig. 7 Total number of transmission slots versus number of attributes

schedule length by 3.12% and 6.56%, as compared to DICA and DICA_EXTENSION, respectively.

The energy consumption [1,20] during the data phase with regard to the number of attributes is shown in Fig. 6. On average, considering all the attributes, FAJIT minimizes the average energy consumption in the data phase by 44.39% and 30.36%, as compared to DICA and DICA_EXTENSION, respectively.

Figure 7 represents the number of transmission slots with regard to the number of attributes. As the number of packets is less for data forwarding, the number of slots required is also less [21]. Thus, FAJIT requires a lesser number of total transmission slots. On average, considering all the attributes, FAJIT minimizes the total number of transmission slots by 13.30% and 8.85%, as compared to DICA and DICA_EXTENSION, respectively.

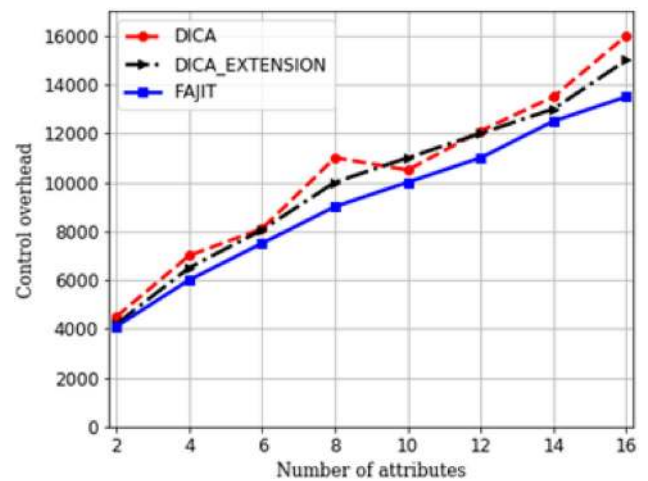


Fig. 8 Control overhead versus Number of attributes

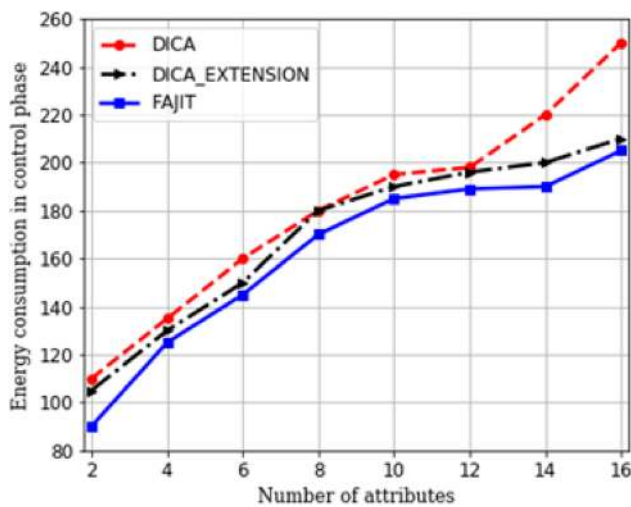


Fig. 9 Energy consumption in the control phase versus number of attributes

The dependency between control overhead and the number of attributes is presented in Fig. 7. The most critical issue of any communication network is the collision, and it is the main challenge of any MAC protocol to avoid it. Usually, in WSNs, several nodes share the same channel. In this paper, scheduling and tree formation are based on labeling and a bottom-up approach, so there is minimal collision chance. During scheduling and tree formation, every node exchanges some control messages with neighbors so that a collision-free schedule can be implemented [24]. These messages constitute control overhead. It is evident from Fig. 8 that if the number of attributes is increased, control overhead also increases. Every node has to select more number of transmission slots [26]. For each slot selection, a number of control messages are exchanged between the given node and its neighbors. Thus, control overhead increases with the number of attributes [18]. As the number of transmission slots needed by FAJIT is less, its control overhead is minimal compared to DICA and DICA_EXTENSION. Energy consumption [19] in the control phase is directly proportional to the control overhead. On average, considering all the attributes, FAJIT minimizes the control overhead to 11%

and 7.71%, as compared to DICA and DICA_EXTENSION, respectively.

Figure 9 depicts that FAJITA energy consumption during the control phase is comparatively lesser than the other algorithms. Energy consumption during the control phase is directly proportional to control overhead. In the FAJIT algorithm, every packet is sent towards a parent to be aggregated as soon as possible. Whereas in DICA-Extension, no such technique is used. Hence, the FAJIT algorithm results in better aggregation compared to DICA-extension. As the aggregation factor increases, nodes send fewer packets. Thus, energy spent in data transmission is also saved. On average, considering all the attributes, FAJIT minimizes the energy consumption to 10.29% and 4.56%, as compared to DICA and DICA_EXTENSION, respectively. The efficiency of FAJIT as compared with DICA and DICA_EXTENSION is summarized in Table 3.

Conclusions and future work

This research work proposes and implements the FAJIT algorithm for single-sink heterogeneous networks for improving energy efficiency in WSN. The idea is to select a different parent for every outgoing packet. Parent selection is done based on the type of packet. It is observed that FAJIT results in better aggregation compared to DICA and DICA_EXTENSION. Thus, it results in smaller schedule length, reduction in energy consumption during the control phase, and data phase. Fuzzification is all about the mapping of each point in the input space with a membership value in a closed unit interval [0, 1]. Min–max normalization is used to calculate membership value. From the demonstrated results, it is evident that FAJIT is improved over traditional DICA and DICA_EXTENSION algorithms. Thus FAJIT seems a better choice for scheduling tree formation in heterogeneous networks. In the proposed algorithms, the first schedule lengths of tentative trees are estimated by the sinks and then tree switching takes place. The other way of handling the problem of schedule length balancing would be to form the trees

Table 3 The efficiency of FAJIT as compared with DICA and DICA_EXTENSION

Performance evaluation metric	The efficiency of FAJIT compared with DICA and DICA_EXTENSION	
	DICA (%)	DICA_EXTENSION (%)
Schedule length	3.12	6.56
Average aggregation factor	6.77	18.22
Control overhead	11	7.71
Average energy consumption in the control phase	10.29	4.56
Average energy consumption in the data phase	44.39	30.36
Number of transmission slots	3.3	8.85

first and then schedule them. Tree switching could be performed after discovering the schedule lengths. Estimation of the new schedule length of a tree could consider the current schedule length.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alsharif MH, Kim S, Kuruolu N (2019) Energy harvesting techniques for wireless sensor networks/radio-frequency identification: a review. *Symmetry* 11(7):865
- Anisi MH, Abdullah AH, Razak SA (2012) Energy-efficient and reliable data delivery in wireless sensor networks. *Wirel Netw* 19(4):495–505
- Anisi MH, Abdullah AH, Coulibaly Y, Razak SA (2013) EDR: efficient data routing in wireless sensor networks. *Int J Ad Hoc Ubiquitous Comput* 12(1):46
- Anisi MH, Abdul-Salaam G, Idris MYI, Wahab AWA, Ahmady I (2017) Energy harvesting and battery power based routing in wireless sensor networks. *Wirel Netw* 23(1):249–266
- Azim MA, Jamalipour A (2007) Performance evaluation of optimized forwarding strategy for flat sensor networks. In: *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference* (pp 710–714), IEEE
- Bagaa M, Younis M, Djenouri D, Derhab A, Badache N (2015) Distributed low-latency data aggregation scheduling in wireless sensor networks. *ACM Trans Sensor Netw* 11(3):1–36
- Balakrishnan M, Johnson EE (2005) Fuzzy diffusion for distributed sensor networks. In: *MILCOM 2005-2005 IEEE Military Communications Conference* (pp 3023–3028), IEEE
- Chithaluru P, Al-Turjman F, Kumar M, Stephan T (2020) I-AREOR: an energy-balanced clustering protocol for implementing green IoT in smart cities. *Sustain Cities Soc* 2020:102254. <https://doi.org/10.1016/j.scs.2020.102254>
- De Guglielmo D, Brienza S, Anastasi G (2016) IEEE 802.15. 4e: a survey. *Comput Commun* 88:1–24
- Fang W, Zhang W, Zhao Q, Ji X, Chen W, Assefa B (2019) Comprehensive analysis of secure data aggregation scheme for industrial wireless sensor network. www.techscience.com. 61(2):583-599
- Fanian F, Rafsanjani MK (2020) A new fuzzy multi-hop clustering protocol with automatic rule tuning for wireless sensor networks. *Appl Soft Comput* 89:106115
- Gherbi C, Aliouat Z, Benmohammed M (2016) An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks. *Energy* 114:647–662
- Halawani S, Khan AW (2010) Sensors lifetime enhancement techniques in wireless sensor networks—a survey. [arXiv:1005.4013](https://arxiv.org/abs/1005.4013)
- Han Z, Wu J, Zhang J, Liu L, Tian K (2014) A general self-organized tree-based energy-balance routing protocol for wireless sensor network. *IEEE Trans Nucl Sci* 61(2):732–740
- Hayouni H, Hamdi M (2017) A data aggregation security enhancing scheme in WSNs using homomorphic encryption. *Intell Autom Soft Comput* 2017:1–9
- Hossein Anisi M, Rezazadeh J, Dehghan M (2008) FEDA: fault-tolerant energy-efficient data aggregation in wireless sensor networks. In: *2008 16th international conference on software, telecommunications and computer networks, Split*, pp 188–192. <https://doi.org/10.1109/SOFTCOM.2008.4669477>
- Hwang RH, Wang CC, Wang WB (2017) A distributed scheduling algorithm for IEEE 802.15. 4e wireless sensor networks. *Comput Stand Interfaces* 52:63–70
- Kang SH, Nguyen T (2012) Distance based thresholds for cluster head selection in wireless sensor networks. *IEEE Commun Lett* 16(9):1396–1399
- Kaswan A, Nitesh K, Jana PK (2017) Energy efficient path selection for mobile sink and data gathering in wireless sensor networks. *AEU-Int J Electron Commun* 73:110–118
- Khan MY, Javaid N, Khan MA, Javaid A, Khan ZA, Qasim U (2013) Hybrid DEEC: towards efficient energy utilization in wireless sensor networks. [arXiv:1303.4679](https://arxiv.org/abs/1303.4679)
- Kim JY, Sharma T, Kumar B, Tomar GS, Berry K, Lee WH (2014) Intercluster ant colony optimization algorithm for wireless sensor network in dense environment. *Int J Distrib Sensor Netw* 10(4):457402
- Kumar H, Singh PK (2018) Comparison and analysis on artificial intelligence based data aggregation techniques in wireless sensor networks. *Procedia Comput Sci* 132:498–506
- Lin Y, Zhang J, Chung HSH, Ip WH, Li Y, Shi YH (2011) An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 42(3):408–420
- Mehmood A, Khan S, Shams B, Lloret J (2015) Energyefficient multilevel and distanceaware clustering mechanism for WSNs. *Int J Commun Syst* 28(5):972–989
- Nehra V, Sharma AK, Tripathi RK (2019) NMR inspired energy efficient protocol for heterogeneous wireless sensor network. *Wirel Netw* 25(6):3689–3700
- Qing L, Zhu Q, Wang M (2006) Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Comput Commun* 29(12):2230–2237
- Rani R, DeeptiKakkar P, Ashish R (2019) Distance based enhanced threshold sensitive stable election routing protocol for heterogeneous wireless sensor network. *Computational Intelligence in Sensor Networks*. Springer, Berlin, Heidelberg, pp 101–122
- Rostami AS, Badkoobe M, Mohanna F, Hosseinabadi AAR, Sangaiha AK (2018) Survey on clustering in heterogeneous and homogeneous wireless sensor networks. *J Supercomput* 74(1):277–323
- Shagari NM, Idris MYI, Salleh RB, Ahmady I, Murtaza G, Shehadeh HA (2020) Heterogeneous energy and traffic aware sleep-awake cluster-based routing protocol for wireless sensor network. *IEEE Access* 8:12232–12252
- Shokouhifar M, Jalali A (2017) Optimized sugeno fuzzy clustering algorithm for wireless sensor networks. *Eng Appli Artif Intell* 60:16–25
- Sia YK, Liew S, Goh HG, Gan M (2014) Spanning multi-tree algorithms for load balancing in multi-sink wireless sensor networks with heterogeneous traffic generating nodes. In: *International conference on frontiers of communications, networks and applications*

- (ICFCNA 2014-Malaysia), Kuala Lumpur, pp 1–6. <https://doi.org/10.1049/cp.2014.1396>
32. Singh SK, Singh MP, Singh DK (2010) Energy-efficient homogeneous clustering algorithm for wireless sensor network. *Int J Wirel Mobile Netw* 2(3):49–61
 33. Soua R, Minet P, Livolant E (2016) Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15. 4e TSCH networks. *Trans Emerg Telecommun Technol* 27(4):557–575
 34. Stephan T, Sharma K, Shankar A, Punitha S, Varadarajan V, Liu P (2020) Fuzzy-logic-inspired zone-based clustering algorithm for wireless sensor networks. *Int J Fuzzy Syst* 2020:1–12
 35. Stephan T, Rajappa A, Sendhil Kumar KS, Gupta S, Shankar A, Vijayakumar V (2020) Modified fuzzy-based greedy routing protocol for VANETs. *J Intell Fuzzy Syst* 2020:1–8
 36. Stephan T, Al-Turjman F, Joseph KS, Balusamy B, Srivastava S (2020) Artificial intelligence inspired energy and spectrum aware cluster based routing protocol for cognitive radio sensor networks. *J Parallel Distrib Comput* 2020:90–105
 37. Upadhyayula S, Gupta SK (2007) Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks. *Ad Hoc Netw* 5(5):626–648
 38. Wang J, Gao Y, Yin X, Li F, Kim HJ (2018) An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. *Wirel Commun Mobile Comput* 2018:9472075
 39. Wang J, Gao Y, Liu W, Sangaiah AK, Kim HJ (2019) An improved routing schema with special clustering using PSO algorithm for heterogeneous wireless sensor network. *Sensors* 19(3):671
 40. Wang J, Gu X, Liu W, Sangaiah AK, Kim H-J (2019) An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Human-centric Comput Inf Sci* 9:1
 41. Wang J, Yang Y, Wang T, Sherratt RS, Zhang J (2020) Big data service architecture: a survey. *J Internet Technol* 21(2):393–405
 42. Wang J, Gao Y, Zhou C, Sherratt S, Wang L (2020) Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs. *Comput Mater Continua* 62(2):695–711
 43. Yu B, Li J (2011) Minimum-time aggregation scheduling in multi-sink sensor networks. In: 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (pp. 422–430), IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.