TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 1710 – 1723
© TÜBİTAK
doi:10.3906/elk-1801-100

Reliable data gathering in the Internet of Things using artificial bee colony

Samad NAJJAR-GHABEL, Shamim YOUSEFI, Leili FARZINVASH*
Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Received: 13.01.2018 • **Accepted/Published Online:** 15.04.2018 • **Final Version:** 27.07.2018

Abstract: The Internet of Things (IoT) technology enables physical devices to communicate with each other for preparing, gathering, and sharing hazard warnings or critical information without human intervention. With respect to emergency applications of IoT technology, an essential issue is to provide an efficient and robust scheme for data gathering. The proposed solution in the existing approaches is to construct a spanning tree over the IoT devices and collect data using the tree. The shortcoming of these algorithms is that they do not take into account the probability of device mobility or failure. In such cases, the spanning tree is split, and it becomes impossible to deliver critical data to the base station on time. In this paper, we propose a reliable spanning tree construction algorithm, which is called reliable spanning tree construction in IoT (RST-IoT). Our algorithm utilizes the artificial bee colony algorithm to generate proper trees. In this method, hop count distances of the devices from the base station, residual energies of the devices, and their mobility probabilities are considered to measure the appropriateness of the trees. Moreover, the proposed algorithm generates a number of trees instead of a single one. These trees are arranged according to their preferences and used for data gathering in succession. Each tree is employed for data gathering upon splitting the preceding one. The simulation results show that RST-IoT improves the reliability of data gathering in emergency applications compared to the previous approaches.

Key words: Internet of things, spanning tree, mobility probability, reliability, residual energy, artificial bee colony

1. Introduction

The Internet of Things (IoT) is a novel paradigm that considers the interaction of smart things. The main idea of IoT is that smart things or objects, such as sensors, actuators, smart phones, and RFID tags, are spread everywhere ubiquitously. These devices have interactions with each other and are able to collect and exchange data. The IoT paradigm covers a broad range of applications, comprising environmental monitoring [1, 2], transportation services, smart cities [3], emergency situation explorers, and industrial processes controllers [4].

Reliable data gathering is an important issue to support emergency applications under IoT technology. The devices send the critical information to the base station for processing and decision making. Furthermore, the base station sends some commands to the devices under its control [5, 6]. Transferring important data over the network requires a reliable backbone. The common approach for data gathering in IoT systems is to construct a spanning tree over the available devices [7–11]. The drawback of the existing algorithms is the unreliability of their generated trees. This shortcoming is mostly due to their failure in considering the characteristics of IoT devices. These devices are usually mobile and equipped with low-power batteries. Therefore, it is possible that the spanning tree is split because of the mobility of some devices or their energy exhaustion.

*Correspondence: l.farzinvash@tabrizu.ac.ir

In this paper, we propose a novel method for constructing spanning trees among IoT devices, namely reliable spanning tree construction in IoT (RST-IoT), which increases the reliability of data gathering. For this purpose, the algorithm takes into account the common features of IoT devices. The considered criteria for tree construction are hop count distances of the devices from the base station and their residual energies. Moreover, as stated in [12], IoT devices have certain mobility patterns over time. Therefore, mobility probabilities of devices are also taken into account in this work. It is worth mentioning that all of the above-mentioned measures are important to construct proper trees. For example, as shown in the simulation results, the resultant spanning trees are more proper when incorporating all of the criteria in comparison with the scenario that only the hop count distance measure is taken into account.

Constructing optimal spanning trees over devices is an NP-hard problem [13]. Swarm intelligence algorithms are appropriate methods in solving such optimization problems [14]. Therefore, we employ the artificial bee colony (ABC) algorithm for tree construction. The considered criteria, comprising hop count distances of the devices from the base station, mobility probabilities of devices, and their residual energies, form the objective of the proposed algorithm. In the resultant trees, the internal nodes are more resident and have more energy. Therefore, the lifetime and the reliability of the obtained trees are increased. Furthermore, the algorithm generates a number of trees instead of a single one. These trees are sorted according to the amount of their properness. First, the most proper tree is employed for data gathering. After splitting this tree, the next most proper tree is used. This procedure continues until all trees are exhausted. Then the algorithm is reexecuted for the next period. In summary, the major contributions of this work are stated as follows:

- We propose a new algorithm for spanning tree construction in IoT systems using the ABC algorithm. Although the problem has previously been studied in [8], it did not address network reliability. Furthermore, the mentioned work proposed heuristic schemes to tackle the problem, which yields less performance in comparison to swarm intelligence-based approaches.

- We enhance the reliability of data gathering using the most effective criteria, comprising the hop count distances of the devices from the base station, mobility probabilities of the devices, and their residual energies. These criteria are employed by the ABC algorithm to construct appropriate trees.

The rest of this paper is organized as follows. Section 2 reviews the related work on the intended problem. The network model is presented in Section 3. In Section 4, the RST-IoT algorithm is discussed, and its features are explained using a practical example. Moreover, we compare the proposed method with previous algorithms in different scenarios in Section 5. Finally, the paper is concluded in Section 6.

2. Related work

The studies related to our work are categorized into two groups. The first group comprises routing algorithms among IoT devices, which have not used a tree structure [12, 15–27]. These approaches considered various measures such as reliability and energy efficiency. The second category is the tree-based schemes, which provided a backbone tree for data transmission to the base station in IoT systems [7–11].

Reliable data routing in IoT was investigated in [12, 15–17]. These works considered different criteria, such as device mobility and link reliability, to select the best routes for data transmission. In [12], the authors studied the mobility of the devices and found that IoT devices have certain mobility patterns over time. Therefore, it is possible to predict the mobility of each device according to its history. Ali et al. [15] designed a stochastic routing algorithm for IoT systems. In this scheme, the authors modeled the network as an absorbing Markov chain to compute the expected delivery ratio and delay. Accordingly, the transmission probability of each link is computed based on the local information. The shortcoming of this approach is the probability of choosing improper paths for data transmission. To provide reliability, the proposed scheme in [16] established backup routes for the nodes. These routes are used for data transmission in the case of primary route failure.

The cross-layer routing scheme was examined in [18], in which different layers cooperate with each other to enhance the performance. This work proposed a mathematical model to select optimal paths among IoT devices. Moreover, physical and link layer parameters such as power levels of the links and the maximum number of required retransmissions are derived from this model. The authors noticed that if the link cost is defined consistent with the objective function, solving the model is equivalent to finding the shortest paths among the devices. Accordingly, they proposed a heuristic algorithm to establish a route between two devices in a short time. In the proposed scheme, the sender initially generates a route request packet. This packet is transmitted toward the base station in a multihop fashion. The base station has some knowledge about the network and selects the least-cost path among the IoT devices. This study focuses on finding the best route and does not adopt backup routes. Therefore, it is vulnerable to device mobility and link failure.

The proposed algorithm in [19] also provided a cross-layer solution for efficient data delivery. The MAC layer is assumed to be TDMA, where a randomization technique is used for scheduling. To this end, the time is divided into a number of frames. The IoT devices compete to get the channel in each frame. This technique diminishes collision considerably. Moreover, it does not require rescheduling if the routing paths are changed. The routing layer of the algorithm employs a bioinspired technique to forward packets toward one of the available base stations. The devices drop some pheromone on the data transmission paths. This behavior constructs pheromone trails, which have the highest concentrations nearby the base stations. To perform data transmission, each device sends its packets to the neighbor with the most pheromone. As the closer devices to the base stations have more pheromone, the packets are delivered to the closest base station after a number of iterations. However, this algorithm is not able to support device mobility due to its slow convergence speed.

Energy-efficient data gathering in IoT systems was examined in [20–25]. Rani et al. [20] proposed a multitier framework, in which the nodes in each tier are responsible for collecting the data of the lower one. The nodes of the first tier monitor the environment and transmit the sensed data to the second one. This procedure continues until all the data reach the uppermost tier, which comprises a number of base stations. To diminish energy consumption, the nodes of the second tier act as cluster heads and aggregate the sensed data. The authors modeled the problem of determining the nodes of each tier as an optimization problem and proved its NP-hardness. Next, they proposed an efficient heuristic algorithm to tackle the problem. To diminish energy consumption, the proposed algorithms in [23–25] exploited the similarity of the sensed data by the sensors. In [23], the nodes send the current sensed data toward the base station if the difference from the previously sent data exceeds a predefined threshold. In the scheme proposed by Jin et al. [24], the correlated data are routed to common relay nodes. This raises the possibility of performing data aggregation. Hence, the amount of transmitted data and the energy consumption are decreased.

Debroy et al. [26] investigated the device-to-device routing problem in cognitive IoT networks. To fairly manage concurrent device-to-device communications, the authors proposed an evolutionary game-based path construction algorithm. The proposed scheme considers the channel availability to maximize the end-to-end data transmission rate. In addition, it adjusts the transmission power in each node to increase the transmission rate while interference is kept at an acceptable level. Zhong et al. [27] also studied cognitive IoT networks. This work exploited opportunistic routing for data transmission. In this scheme the forwarder nodes are selected dynamically based on a number of criteria per packet. The considered measures for forwarder node selection in [27] are channel availability and the amount of required energy for data transmission.

In the rest of this section, we analyze the existing tree construction schemes for IoT systems. In the proposed algorithm in [8], the spanning tree is constructed in some rounds. Initially, the spanning tree contains only the base station. The base station broadcasts control packets to its neighbors. These nodes select the base station as their parent. In the next round, IoT devices repeat the same procedure and announce their presence in the tree by broadcasting control packets. The nontree devices that receive these packets select their parents and join the tree. This procedure continues until the tree is completed. It is possible that a nontree device receives two or more control packets. In this case, the most appropriate device is selected as the parent. The measures to evaluate the properness of a parent are the number of children, residual energy, and hop count distance from the base station. More specifically, it is preferred to select a parent with fewer children and more residual energy that is closer to the base station.

Tang et al. [9] proposed a hierarchical algorithm for spanning tree construction. In this scheme, the network is partitioned into a number of cells. Next, some cells are grouped into a subregion. The upper-level subregions are formed similarly. This procedure continues until the hierarchical tree is completed. This approach is developed to support spatial range queries. To diminish the energy consumption, each device reports its sensed data to the base station only if its difference from the previously reported data is discernible. The proposed algorithm in [10] also constructed a hierarchical spanning tree for processing spatial range queries. These algorithms were aimed at efficient processing of spatial range queries while failing to consider network reliability. Li et al. [11] also employed a spanning tree for data collection. This work assumed that the tree is given in advance, and its concern was to provide a proper data transmission schedule such that delay requirement is preserved. In some applications, it is sufficient to disseminate messages to a specific subset of nodes. In this case, a multicast tree is constructed among the base station and the mentioned nodes [28].

3. The proposed model

The underlying IoT is modeled as an undirected graph $G = (V, E)$. In the intended setting, V represents the set of the network nodes comprising devices and the base station, and E denotes the set of the links between the nodes. Furthermore, each node n_i has two main properties:

- p_i: IoT devices are almost moving randomly. The mobility probability of the devices can be predicted according to their histories [12]. The mobility probability of n_i, namely p_i, is a random number within the range of [0,1]. The mobility probability of the base station is set to 0. In other words, it is assumed that this node has no mobility.

- e_i: This variable shows the residual energy of n_i. The energy of the base station is assumed to be unlimited.

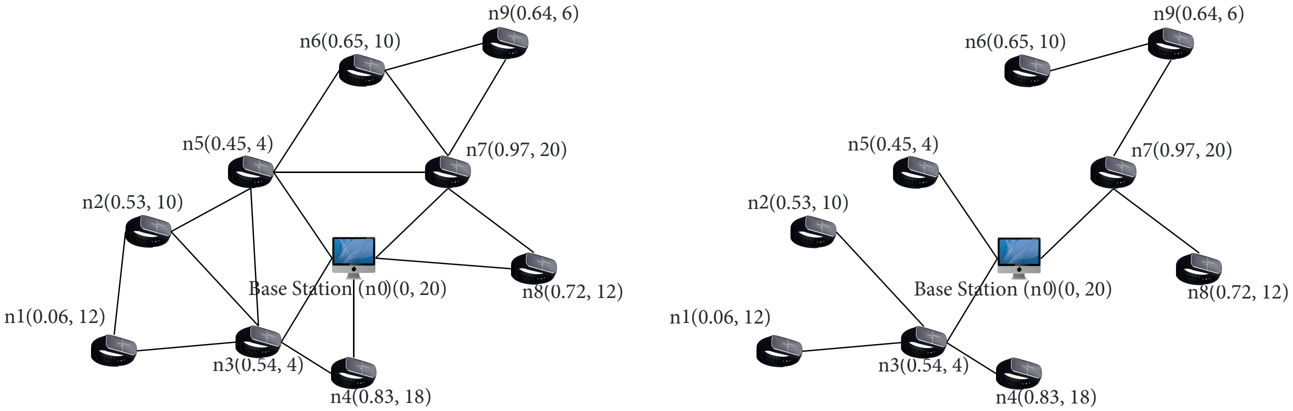
In the proposed algorithm, it is assumed that K spanning trees are constructed over the network. Spanning tree t_k has the following features, which are used in the proposed algorithm:

- mp_k: As stated before, the mobility probability of internal nodes is an effective criterion on the reliability of spanning trees. In the proposed algorithm, the most resident devices are selected as the internal nodes in order to increase the reliability of the trees. Each internal node n_i has some children over t_k, where the number of children is shown by ch_i^k. The internal nodes with more children play more important roles in the data gathering procedure. Therefore, the number of children of the internal nodes should be considered as an effective factor. Accordingly, the mobility probability of t_k (i.e., mp_k) is defined as:

$$mp_k = \sum_{n_i \in V} (ch_i^k + 1)\, p_i. \tag{1}$$

To clarify the concept of mp-variable, in the following we compute mp of the proposed example in Figure 1a. A sample spanning tree for this configuration is given in Figure 1b. Using Eq. (1), the mp-variable of this tree is derived as follows:

$$mp_k = 4\,p_0 + p_1 + p_2 + 4\,p_3 + p_4 + p_5 + p_6 + 3\,p_7 + p_8 + 2\,p_9 = 9.68. \tag{2}$$



(a) The intended network.

(b) A sample spanning tree.

Figure 1. An example IoT network. The p- and e-variables of the nodes are illustrated in the figure.

- er_k: The other important factor to increase the robustness of the spanning trees is to select the nodes with more energy as the internal nodes. Considering the above discussion about the mp-variable, the ch-variables of the internal nodes are effective on the reliability of the spanning trees. Therefore, the residual energy of t_k, namely er_k, is defined as:

$$er_k = \sum_{n_i \in V} (ch_i^k + 1)\, e_i. \tag{3}$$

The er of the depicted tree in Figure 1b is computed in Eq. (4):

$$er_k = 4\,e_0 + e_1 + e_2 + 4\,e_3 + e_4 + e_5 + e_6 + 3\,e_7 + e_8 + 2\,e_9 = 214\,J. \tag{4}$$

- hc_k: This criterion presents the total hop count distances of the devices from the base station and is formulated as:

$$hc_k = \sum_{n_i \in V} h_i^k, \tag{5}$$

where h_i^k denotes the hop count distance of node n_i from the base station over t_k. Although this measure does not impact reliability, it is important to reduce the data collection delay and improve the lifetime. Therefore, this criterion is also considered in the tree construction algorithm.

The hc of the illustrated tree in Figure 1b is equal to 16 and is calculated as follows:

$$hc_k = h_1^k + h_2^k + h_3^k + h_4^k + h_5^k + h_6^k + h_7^k + h_8^k + h_9^k = 16. \tag{6}$$

The utilized notations in the intended model are listed in Table 1.

Table 1. The list of utilized notations in this paper.

Definitions	Symbols
V	Set of the nodes comprising IoT devices and base station
E	Set of the network links
n_i	The ith node of V
p_i	Mobility probability of n_i
e_i	Residual energy of n_i
K	Number of spanning trees
t_k	The kth spanning tree
ch_i^k	Number of children of n_i over t_k
mp_k	Total mobility probability of t_k
er_k	Total residual energy of t_k
h_i^k	Hop count distance of n_i from the base station over t_k
hc_k	Total hop count distance of t_k
$\|.\|$	Size of a given set

4. Reliable tree-based data gathering in IoT

In this section, we explain the proposed approach for reliable tree construction in IoT systems. The algorithm constructs a set of reliable spanning trees for data gathering. First, data gathering is performed over the most proper tree. This tree remains operational until one of its internal devices moves or runs out of energy. In this case, the tree is split and data gathering is interrupted. Therefore, the next most proper spanning tree is employed. This procedure continues until all trees are utilized. The rest of this section is organized as follows. We briefly explain the ABC algorithm in Section 4.1. Next, the RST-IoT algorithm is expounded using an illustrative example in Section 4.2.

4.1. ABC algorithm

In the proposed algorithm, we employ the ABC algorithm for tree construction. The advantage of this method is that it does not require parameter setting. Moreover, it has been shown that this approach yields better performance in comparison to other swarm intelligence algorithms such as the genetic algorithm (GA) and particle swarm optimization (PSO) [29, 30]. This optimization algorithm is based on the intelligent behavior of bees for food collection. A bee colony contains three types of bees:

- Employed bee: An employed bee is currently employed at a specific food source to exploit it.
- Onlooker bee: The bees that find a proper food source based on the information given by the employed bees.

- Scout bee: These bees search the environment randomly to find a new food source.

The ABC algorithm is designed according to the communications of the bees to find the best food sources. In this algorithm, the position of a food source denotes a possible solution of the optimization problem. Moreover, the nectar amount of a food source is equivalent to its quality. The steps of the algorithm are briefly described as follows:

1. Initialization: In order to solve the optimization problems using the ABC algorithm, an initial population is generated randomly. Each member of this population is a possible solution (food source position) for the optimization problem. Moreover, the number of employed bees is equal to the size of the population.

2. Employed bee phase: After generating the initial population, the food source positions (solutions) are updated in some rounds. An employed bee modifies the position of a food source (solution) in her memory using the local information and checks the nectar amount (quality) of the new source (new solution). If the nectar amount increases by this modification, the bee remembers the new position (new solution).

3. Onlooker bee phase: The employed bees share the information of the food sources, comprising their positions and nectar amounts, with the onlooker bees. Each onlooker bee selects a food source with a probability, which is computed using the nectar amounts of the sources. Similar to the previous phase, each onlooker bee makes a modification of the position of the selected food source (solution), considering the local information, and tests the nectar amount (quality) of the new source (solution). If the nectar amount of the new source is more than that of the old one, the bee remembers the new position (new solution).

4. Scout bee phase: The scout bees search the environment to find new food sources (solutions) randomly and substitute them for the abandoned ones. In other words, these bees replace a predetermined number of food sources (solutions) with worse nectar amounts by random new food sources (solutions).

4.2. RST-IoT algorithm

Algorithm 1 illustrates the proposed algorithm, which customizes the ABC algorithm to solve the spanning tree construction problem. In the proposed scheme, K random spanning trees are constructed in the initialization phase. The other phases are repeated with $maxIT$ iterations to find proper trees. In this context, cIt displays the number of the current iteration. The resultant trees of each phase form the current solution space, which is denoted by csl. The final solution space, namely fsl, is derived after the execution of the algorithm. Moreover, the nectar amount of t_k is shown by nc_k, and NC is a vector of size K that maintains nc-variables of the trees. The steps of the proposed algorithm can be explained as follows:

1. Initialization: In this phase, K spanning trees are constructed over the network. These trees form the initial solution space of the ABC, which is denoted by isl. Tree t_k is modeled using a binary array of length $|E|$. The corresponding elements to the tree links are set to 1. To construct t_k, $|V| - 1$ elements of the array are randomly chosen and set to 1. Other elements are filled with 0, meaning that their corresponding links are not included in t_k. In the proposed scheme, it is possible that the selected links form a loop. Therefore, the procedure should be repeated until a tree is obtained.

2. Employed bee (first phase of Algorithm 1): After generating initial trees, the employed bees reorganize the trees in the hope of finding more proper ones. To update t_k, an employed bee modifies its corresponding

array and exchanges a pair of 0- and 1-value elements. The bee remembers the new tree if its nectar amount is more than the old one. Similar to the initialization phase, tree reorganization may lead to loop formation. Therefore, the employed bees should repeat tree reorganization until no loop exists in the new solution.

3. Onlooker bee (second phase of Algorithm 1): The aim of employing onlooker bees is to improve the quality of worse trees in a probabilistic manner. In other words, the trees with less nectar should be selected with a higher probability. Therefore, the probability of selecting t_k, namely sel_k, is defined as:

$$sel_k = \frac{sb_k}{\sum_{m=1}^{K} sb_m}, \tag{7}$$

where sb_k denotes the suitability of t_k and is computed as:

$$sb_k = \sum_{m=1}^{K} nc_m - nc_k. \tag{8}$$

Each onlooker bee adopts a spanning tree using a roulette wheel selection (RWS) mechanism. The onlooker bee modifies the selected tree with the hope of increasing its nectar amount. For this purpose, it selects a random pair of 0- and 1-elements from the corresponding array to the tree and flips their values. The onlooker bee remembers the new tree as the solution if it increases the nectar amount compared to the old one. It is worth mentioning that the loop preservation condition is checked after applying the modifications. If a loop is formed as a result of tree reorganization, the onlooker bee repeats the mentioned operation until a loop-free tree is established.

4. Scout bee (third phase of Algorithm 1): Some trees do not have enough nectar and should be abandoned. Therefore, in this phase, the scout bees consider the nectar amounts of the trees and replace the trees with low nectar amounts with new random ones.

```
Input: Graph $G = (V, E)$.
Output: $fsl$.
$[isl] = Initialization(G)$.
$cIt \leftarrow 0$.
$csl \leftarrow isl$.
for $cIt = 1$ to $maxIt$ do
    $[csl, NC] = Employed\,bee(G, csl, NC)$.
    $[csl, NC] = Onlooker\,bee(G, csl, NC)$.
    $[csl, NC] = Scout\,bee(G, csl, NC)$.
end for
Sort the trees in $csl$ according to their $nc$-variables.
$fsl \leftarrow csl$.
```

Algorithm 1: Spanning tree construction.

The remaining point is to compute nc_k. To solve multiobjective problems with the ABC algorithm, the weighted sum of the chosen criteria is considered as the nectar amount of a given solution. In the intended problem, our aim is to increase the reliability of the trees. Spanning tree t_k is split if its internal nodes do not work properly, which occurs if these nodes move or their energies are exhausted. Therefore, the criteria to evaluate the appropriateness of t_k are mp_k, er_k, and hc_k. More specifically, nc_k is defined as the linear combination of these criteria in such a way that mp_k and hc_k are minimized, while er_k remains as high as possible. This variable is described as follows:

$$nc_k = w_1\left(1 - \frac{mp_k - mpMin}{mpMax - mpMin}\right) + w_2\left(\frac{er_k - erMin}{erMax - erMin}\right) + w_3\left(1 - \frac{hc_k - hcMin}{hcMax - hcMin},\right)$$

$$\begin{aligned} &where\\ &\quad mpMax = max_{1 \leq k \leq K}\, mp_k,\\ &\quad mpMin = min_{1 \leq k \leq K}\, mp_k,\\ &\quad erMax = max_{1 \leq k \leq K}\, er_k,\\ &\quad erMin = min_{1 \leq k \leq K}\, er_k,\\ &\quad hcMax = max_{1 \leq k \leq K}\, hc_k,\\ &\quad hcMin = min_{1 \leq k \leq K}\, hc_k. \end{aligned} \tag{9}$$

In this equation, $mpMin$, $mpMax$, $erMin$, $erMax$, $hcMin$, and $hcMax$ are used to normalize the phrases and keep them at the same level. In addition, w_1, w_2, and w_3 are the corresponding weights to mobility probability, residual energy, and hop count distance measures.

To clarify the proposed algorithm, we apply it to the presented tree in Figure 1b. The corresponding array to this tree is depicted in Figure 2. The elements of this array indicate the presence of links (n_1, n_2), (n_2, n_5), (n_3, n_5), (n_2, n_3), (n_1, n_3), (n_0, n_3), (n_3, n_4), (n_0, n_4), (n_5, n_6), (n_6, n_9), (n_7, n_9), (n_6, n_7), (n_0, n_7), (n_7, n_8), (n_0, n_8), (n_0, n_5), and (n_5, n_7) in the network. The amounts of corresponding elements to the depicted tree in Figure 1b are set to 1. The nectar amount of the considered tree is equal to 0.46.

0	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0

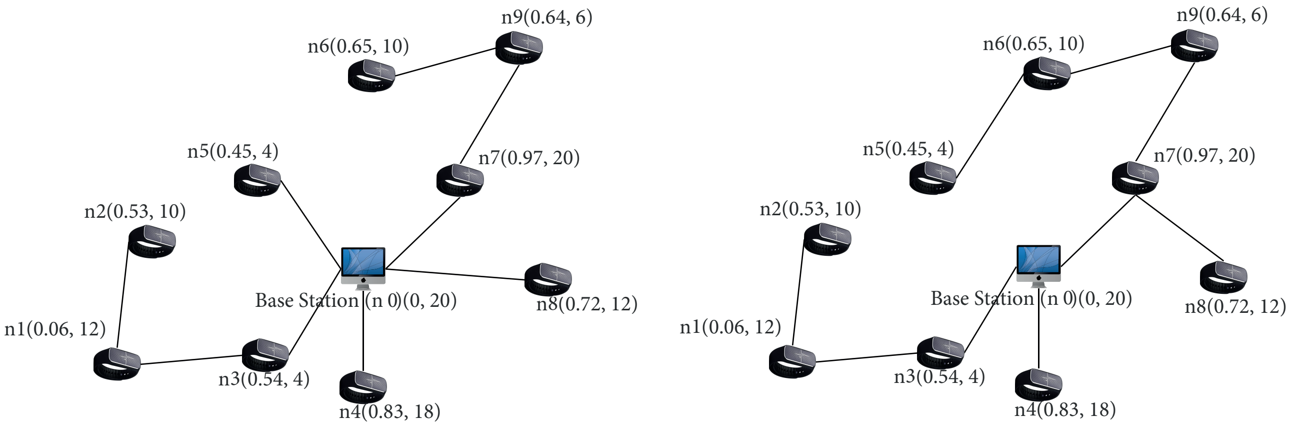
Figure 2. The corresponding array to the given tree in Figure 1b.

After the accomplishment of the tree construction procedure, a number of trees are derived. Figure 3 presents two most suitable trees for the proposed configuration in Figure 1a.

5. Performance analysis

In this section, we compare the achieved performance by RST-IoT against two schemes. The first one is the proposed algorithm in [8], which is called ETSP. In the second approach, named HCT-IoT, we modify our algorithm and only consider the hop count distance as the performance measure. As RST-IoT and HCT-IoT employ the ABC algorithm, the resultant trees may differ in various runs. To reduce the variance of the results, these algorithms are executed 50 times and the average is considered as the outcome. The algorithms are implemented using MATLAB.

The dimensions of the simulation environment are $120\,m \times 120\,m$, where 20 devices are scattered randomly over the monitoring area. The devices are assumed to be heterogeneous; that is, their initial energies and mobility probabilities are not identical. The mentioned criteria are selected within the ranges of $[1\,J - 20\,J]$ and $[0 - 1]$, respectively. To implement node mobility, the time domain is divided into equal-sized slots. In



(a) The most suitable tree.

(b) The second most suitable tree.

Figure 3. The generated spanning trees by RST-IoT for the given network in Figure 1a.

each time slot, node n_i moves with probability p_i. Moreover, the transmission ranges of all devices are set to $10\,m$. Table 2 lists the amounts of utilized parameters.

Table 2. The amounts of employed parameters.

Parameter	Value
Network dimensions	$120\,m \times 120\,m$
Transmission range	$10\,m$
Number of employed bees	K
Number of onlooker bees	$K/4$
Number of scout bees	K
w_1	0.45
w_2	0.45
w_3	0.1
$maxIt$	1000

To investigate the effectiveness of the proposed algorithm, we consider two measures of reliability and energy consumption. These criteria are defined as follows:

- Reliability is considered as a property of IoT systems that enables such networks to remain operational in the case of mobility or energy exhaustion of the devices. This criterion is formally defined as:

$$Reliability = \frac{DFF}{TDFF}, \tag{10}$$

  where DFF represents the number of devices that fail simultaneously due to mobility or energy exhaustion so that the spanning tree is not split. Obviously, such failures occur at the leaf nodes. The TDFF measure represents the total number of failures. The resultant reliability by the considered algorithms is studied in Section 5.1.

- Average energy consumption is the average of the energy consumed by devices to transmit the generated data to the base station. We study this measure in Section 5.2. To have an accurate comparison of the methods, it is assumed that no fault occurs in the simulations performed in that section.

5.1. Reliability comparison

An important issue in IoT systems is the reliability of data gathering. In the simulations performed in this section, we compare RST-IoT to HCT-IoT and ETSP from this point of view. Figure 4 illustrates the reliability of the considered algorithms by varying the number of failures from 1 to 4. The number of nodes is fixed at 20 in this figure. In addition, parameter K is assumed to be equal to 10. The reported results in this figure show that RST-IoT is more reliable than the other algorithms. More specifically, it improves the reliability by 24.23% and 55.78% when compared to HCT-IoT and ETSP, respectively.

The impact of enlarging the IoT system on its reliability is investigated in Figure 5. In the reported simulations in this figure, K is set to 20, and the number of nodes is varied from 4 to 20. From this figure, we can see that network enlargement diminishes the reliability. Furthermore, according to the results in this figure, RST-IoT improves the reliability by 46% and 82% on average compared to HCT-IoT and ETSP, respectively.

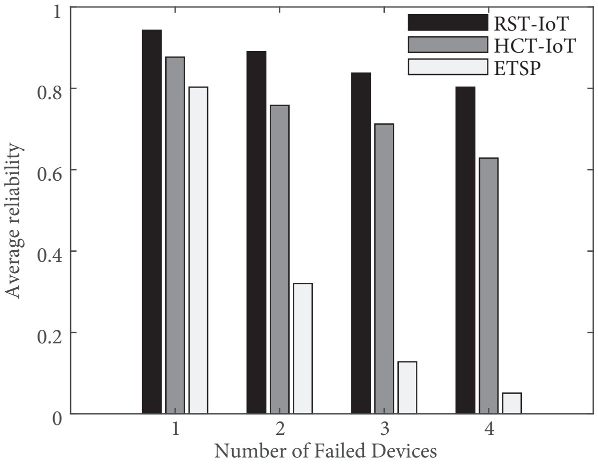


Figure 4. The achieved reliability versus number of simultaneous failures.

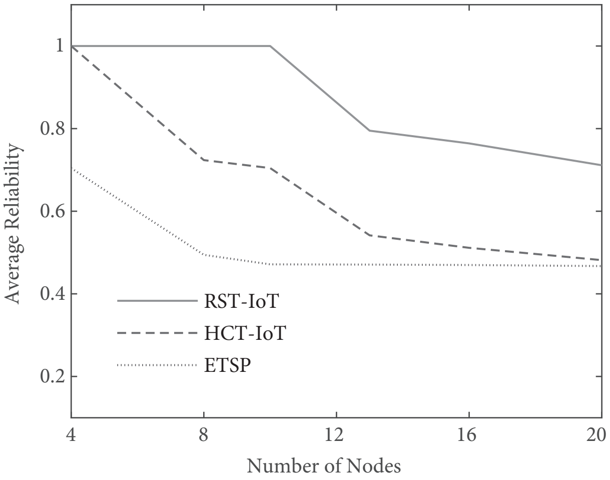


Figure 5. The achieved reliability versus number of nodes.

The next considered criterion is the number of spanning trees. We vary the number of spanning trees from 1 to 20 to investigate its impact on the reliability. The results are reported in Figure 6. The number of nodes is set to 20 in the reported results in this figure. From the findings of this set of simulations, we can see that the average reliability is improved by increasing the number of trees. These results indicate that RST is more fault tolerant than HCT-IoT and improves reliability by 20% relative to this algorithm. Since ETSP generates only one spanning tree, it is not considered in this experiment.

The time complexity of the performed simulations in Figures 5 and 6 are reported in Tables 3 and 4, respectively. As is expected, the running time of the considered algorithms increases with enlarging node set or increasing parameter K. In addition, it is seen from these tables that the running time of the proposed algorithm is acceptable. Hence, it can be employed in practical scenarios.

5.2. Consumed energy comparison

Consumed energy of the IoT system is defined as the total energy consumed by all devices. Simulation results in Figure 7 reveal that the consumed energy by HCT-IoT and ESTP is more than the energy required by RST-IoT. This method improves average consumed energy in the data gathering process by incorporating the residual energy and hop count measures in the objective function. Furthermore, the obtained results illustrate

Table 3. The running time of the simulations in Figure 5 (seconds).

Algorithm / $\|V\|$	4	8	12	16	20
RST-IoT	1.98	2.27	2.29	3.13	4.20
HCT-IoT	1.92	2.14	2.19	3.01	3.96
ETSP	0.05	0.06	0.08	0.15	0.18

Table 4. The running time of the simulations in Figure 6 (seconds).

Algorithm / K	1	5	10	15	20
RST-IoT	2.42	3.067	4.20	5.21	5.60
HCT-IoT	2.28	2.83	3.96	5.10	5.53

that RST-IoT decreases the average consumed energy by 45.02% and 51.13% in comparison to HCT-IoT and ETSP, respectively. Based on Figure 7, we can also infer that the consumed energy by HCT-IoT is lower than the energy required for ETSP. This is because it diminishes total hop count distance, which leads to energy consumption reduction. Moreover, the performance gap between our scheme and other methods is increased by enlarging the network. This verifies the scalability of our algorithm in comparison to HCT-IoT and ETSP. The running time of the presented simulations in Figure 7 is the same as the reported results in Table 3.

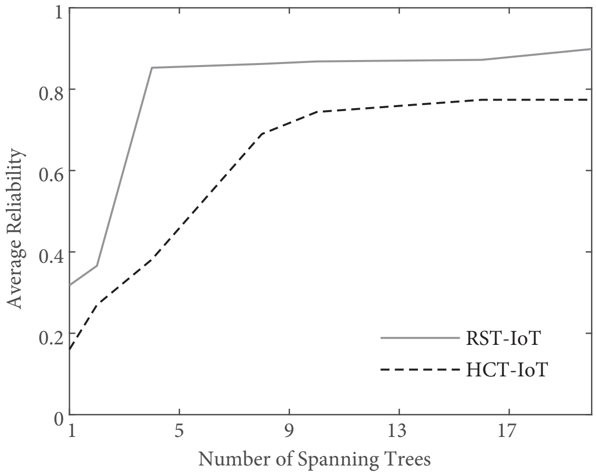


Figure 6. The achieved reliability versus number of spanning trees.

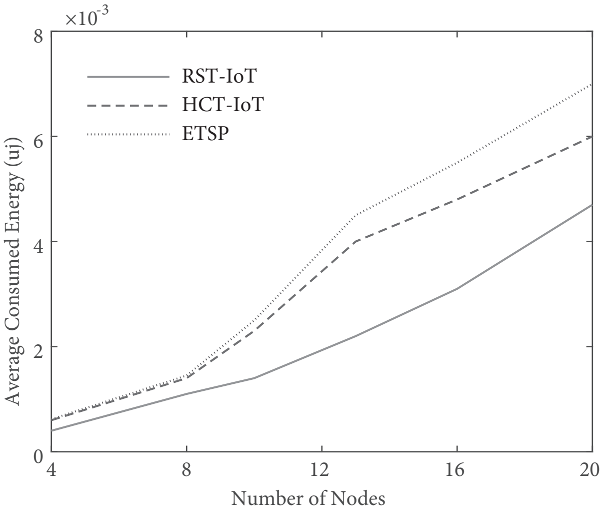


Figure 7. Average consumed energy versus number of nodes.

6. Conclusion

In this study, we addressed the problem of reliable data gathering in IoT systems. Our proposed algorithm, namely RST-IoT, utilizes a tree structure for data collection. To achieve high-throughput solutions, we customized the ABC algorithm for the tree construction problem. The advantage of using this technique is that it is a swarm intelligence optimization scheme and generates near-optimal trees. The nectar amount of each solution is computed using a number of criteria, comprising hop count distances between devices and the base station, residual energies of the devices, and their mobility probabilities. Through the use of the ABC, a number of spanning trees are generated. These trees are sorted according to their preferences, and each tree is employed for data gathering after the splitting of the preceding one. The simulation results indicate that RST-IoT is superior to the existing approaches in terms of reliability and energy consumption. Therefore, it is suitable to handle emergency applications under IoT technology.

References

[1] Alaa M, Zaidan AA, Zaidan BB, Talal M, Kiah MLM. A review of smart home applications based on Internet of Things. J Netw Comput Appl 2016; 97: 48-65.

[2] Perumal T, Sulaiman MN, Datta SK, Leong CY. Rule-based conflict resolution framework for internet of things device management in smart home environment. In: IEEE Global Conference on Consumer Electronics; 11–14 October 2016; Kyoto, Japan.

[3] Li X, Lu R, Liang X, Lin X. Smart community: an Internet of Things application. IEEE Commun Mag 2011; 49: 68-75.

[4] Whitmore A, Agarwal A, Xu LD. The Internet of Things: a survey of topics and trends. Inform Syst Front 2015; 17: 261-274.

[5] Xu LD, He W, Li S. Internet of Things in industries: a survey. IEEE T Ind Inform 2014; 10: 2233-2243.

[6] Bello O, Zeadally S. Intelligent device-to-device communication in the Internet of Things. IEEE Syst J 2014; 10: 1172-1182.

[7] Meghanathan N. Link selection strategies based on network analysis to determine stable and energy-efficient data gathering trees for mobile sensor networks. Ad Hoc Netw 2017; 62: 50-75.

[8] Qiu T, Liu X, Feng L, Zhou Y, Zheng K. An efficient tree-based self-organizing protocol for Internet of Things. IEEE Access 2016; 4: 3535-3546.

[9] Tang J, Zhou ZB, Niu J, Wang Q. An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of Things. J Netw Comput Appl 2014; 40: 1-11.

[10] Zhou ZB, Tang J, Zhang LJ, Ning K, Wang Q. EGF-tree: an energy-efficient index tree for facilitating multi-region query aggregation in the Internet of Things. Pers Ubiquit Comput 2014; 18: 951–966.

[11] Li Z, Zhang W, Qiao D, Peng Y. Lifetime balanced data aggregation for the Internet of Things. Comput Electr Eng 2017; 58: 244-264.

[12] Huang J, Wang S, Cheng X, Bi J. Big data routing in D2D communications with cognitive radio capability. IEEE Wirel Commun 2016; 23: 45-51.

[13] Custic A, Zhang R, Punnen AP. The quadratic minimum spanning tree problem and its variations. Discrete Optim 2018; 27: 73-87.

[14] Moharam R, Morsy E. Genetic algorithms to balanced tree structures in graphs. Swarm Evol Comput 2016; 32: 132-139.

[15] Ali Z, Abbas ZH, Li FY. A stochastic routing algorithm for distributed IoT with unreliable wireless links. In: IEEE Vehicular Technology Conference; 15–18 May 2016; Nanjing, China.

[16] Gia TN, Rahmani AM, Westerlund T, Liljeberg P, Tenhunen H. Fault tolerant and scalable IoT-based architecture for health monitoring. In: IEEE Sensors Applications Symposium; 25 June 2015; Zadar, Croatia.

[17] Rani S, Ahmed SH, Talwar R, Malhotra J, Song H. IoMT: a reliable cross layer protocol for Internet of Multimedia Things. IEEE Internet Things 2017; 4: 832-839.

[18] Han C, Jornet JM, Fadel E, Akyildiz IF. A cross-layer communication module for the Internet of Things. Comput Netw 2013; 57: 622-633.

[19] Goins JB, Roy AG, Mayaram K, Fiez TS. JetNet: a proposed protocol for reliable packet delivery in low-power IoT applications. In: IEEE World Forum on Internet of Things; 12–14 December 2016; Reston, VA, USA.

[20] Rani S, Talwar R, Malhotra J, Ahmed SH, Sarkar M, Song H. A novel scheme for an energy efficient Internet of Things based on wireless sensor networks. Sensors 2015; 15: 28603-28626.

[21] Liang JM, Chen JJ, Cheng HH, Tseng YC. An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for Internet of Things. IEEE J Em Sel Top C 2013; 3: 13-22.

[22] Shrouf F, Ordieres J, Miragliotta G. Smart factories in Industry 4.0: a review of the concept and of energy management approached in production based on the Internet of Things paradigm. In: IEEE Industrial Engineering and Engineering Management; 9–12 December 2014; Bandar Sunway, Malaysia.

[23] Alduais NAM, Abdullah J, Jamil A, Audah L. An efficient data collection and dissemination for IoT based WSN. In: IEEE Information Technology, Electronic and Mobile Communication Conference; 13–15 October 2016; Vancouver, Canada.

[24] Jin Y, Gormus S, Kulkarni P, Sooriyabandara M. Content centric routing in IoT networks and its integration in RPL. Comput Commun 2016; 89-90: 87-104.

[25] Kim DY, Kim S. Efficient path selection for IoT devices in heterogeneous service environments. Lect Notes Electr En 2016; 421: 149-153.

[26] Debroy S, Samanta P, Bashir A, Chatterjee M. SpEED-IoT: spectrum aware energy efficient routing for device-to-device IoT communication. Future Gener Comp Sy 2018 (in press).

[27] Zhong X, Lu R, Li L, Zhang S. ETOR: energy and trust aware opportunistic routing in cognitive radio social Internet of Things. In: IEEE Global Communications Conference; 4–8 December 2017; Marina Bay Sands, Singapore.

[28] Pan MS, Yang SW. A lightweight and distributed geographic multicast routing protocol for IoT applications. Comput Netw 2017; 112: 95-107.

[29] Akay BB, Karaboga D. Artificial bee colony algorithm variants on constrained optimization. International Journal of Optimization and Control: Theories & Applications 2017; 7: 98-111.

[30] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput 2009; 214: 108-132.