Journal of Applied Research and Technology

www.jart.ccadet.unam.mx
Journal of Applied Research and Technology 16 (2018) 524-543

Original

Lifetime enhancement in wireless sensor networks using binary search tree based data aggregation

Gopikrishnan S. [a,*], Priakanth P. [b]

[a] *Assistant Professor, Department of Information Technology, Karpagam College of Engineering, Coimbatore, India..*

[b] *Professor, Department of Computer Technology, Kongu Engineering College, Erode, India.*

Received dd mm aaaa; accepted dd mm aaaa
Available online dd mm aaaa

Abstract: Energy efficient data aggregation is a key solution to enhance the lifetime of wireless sensor networks since sensor nodes are battery-powered and deployed in remote environments. This article explore a two-hop data aggregation tree construction algorithm using binary search tree to reduce the total energy consumption of sensor nodes in wireless sensor networks. An adaptive and hybrid routing algorithm for simultaneous data aggregation and exploit the data correlation between nodes using the two-hop data aggregation tree framework is proposed. Routes are chosen based on the shortest response time for the broadcasting request to minimize the total energy expended by the network. This paper also proposes a high secure asymmetric key cryptography algorithm to provide the secure data communication among the network. The data aggregation function that is used in the proposed routing algorithm enhances the lifetime of sensor network by resolving the delay, collision and security issues. Simulations results show that the binary tree based data aggregation can appreciably reduces the total energy consumption and resolves the maximum data aggregation issues in wireless sensor network.

Keywords: Energy Efficiency, Data Aggregation, Delay, Collision, Security, Wireless Sensor Network

1. INTRODUCTION

WSNs are the deployment of tiny sensor nodes over an area to monitor continuously variable physical phenomena, such as temperature, pressure and humidity. WSNs play a significant role in a variety of appliances and applications. Numerous disciplines use WSNs for different applications, such as monitoring specific features or targets, especially in rescue, surveillance, medical, engineering and industrial applications. They can also be deployed in underground and underwater locations, as well as the normal landscape.

Sensor nodes are normally deployed in an ad-hoc manner, are operated in a distributed way and coordinate with each other to fulfill a common task. Sensor nodes have reduced computation and communication capabilities and are usually non-rechargeable. Depending on the application, there are several kinds of reporting methods for WSNs, such as periodical reporting, reporting by request and event-driven reporting. The advances in WSNs show potential, as well as pose challenges, such as resource limitations, dynamic environments and various application needs. These challenges and trade-offs also

* Corresponding author.
E-mail address: gopikrishnanme@gmail.com (Gopikrishnan S.).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

http://

include data aggregation issues, such as aggregation delay, collision, security and energy.

1.1 PROBLEM STATEMENT AND OBJECTIVES

Data aggregation is one of the main functionalities of sensor nodes. The sensors normally sense the data from the environment and transmit them to the base station or sink node. The regularity of sending the data and the number of sensors that send the data depend on the particular application. Data aggregation is outlined as the method of aggregating the data from multiple sensors in order to abolish redundant transmission and process the data about the sensed environment, after which combined information to the base station is provided. Data aggregation needs a technique for combining the sensed data into high-quality transferable information. It involves aggregation algorithms, which collect the sensed data from multiple sensors nodes and transmit the data to the base station for further processing. The various design issues of the data aggregation algorithm are discussed here. By resolving these issues, an aggregation algorithm can be implemented in all applications.

1) Since the sensor networks are employed in an unattended environment, they are naturally untrustworthy, while assured information could also be unavailable. The number of nodes present in the network and the number of nodes that are employed in environmental monitoring also make it difficult to correctly and completely acquire data of the sensor nodes.

2) Ensuring that the sensor nodes consume less energy should be achieved by transmitting the data directly to the base station or reducing data transmission to the base station.

3) Eliminating the transmission of redundant data should be carried out in order to facilitate data reduction algorithms and avoid the retransmission of a successful transmission.

4) The delay in data aggregation should be reduced to conserve energy in the sensors.

Applying the lightweight cryptography algorithm to secure the network communication is preferable when the sensor network is deployed through event-driven applications.

1.2 CONTRIBUTIONS MADE IN THIS ARTICLE

This article proposes a hybrid communication for energy-efficient data (HCED) aggregation in WSNs. The proposed algorithm provides energy-efficient data aggregation among the sensor nodes. This algorithm has three parts: i) an energy-efficient aggregation tree construction, ii) a collision-free data aggregation algorithm and iii) an asymmetric key cryptography algorithm for secure data aggregation.

The first phase proposes an energy-efficient tree construction algorithm based on the binary search tree model. It follows a two-hop routing model for data aggregation. To resolve the energy-related issues, most researchers have proposed the distributed algorithm for data aggregation. Therefore, to resolve the energy and delay issues, the research have been determined to design distributed tree-based data aggregation. Although many distributed solutions that have been proposed for cluster-based data aggregation, collision issues have been reduced on cluster-head selection. Since each node in a binary tree model has a maximum of two nodes, which reduce collision issues, using the binary tree model is highly recommended.

In second phase, a hybrid delay-efficient data aggregation algorithm, which reduces the energy consumption by performing fast data aggregation is proposed. This hybrid data aggregation algorithm is a combination of distributed and centralized aggregation mechanisms to perform simultaneous data aggregation from sensor nodes.

Finally, secured data aggregation has been achieved through an asymmetric cryptography technique proposed in HCED. The proposed encryption algorithm reduces the computation overhead of the network more than the existing models.

The rest of this paper is organized as follows. Section 2 briefly describes some of the existing schemes that resolve data aggregation issues for WSNs. Section 3 provides the details about the network models and assumptions made regarding this proposed model. A brief discussion on the proposed HCED approach is presented in Section 4. Section 5 discusses how the HCED scheme will resolve the maximum data aggregation issues in an energy-efficient way. The simulation results and performance analysis of the proposed algorithm is discussed in Section 6. Finally, Section 7 provides the conclusion along with the contributions made in this paper.

2. RELATED STUDIES

Over the last decade, many data aggregation schemes have been proposed for energy efficiency and scalable data

collection operations in WSNs. Many related articles have been studied and analyzed in terms of the pros and cons regarding the various data aggregation issues, as discussed in 1.1. Based on the data aggregation protocols, the existing models can be classified into two categories: tree-based and cluster-based data aggregation. This section presents a review of existing data aggregation schemes, which can be classified as centralized, distributed and hybrid approach-based schemes. These classifications are related to tree-based data aggregation protocols. Most of the existing solutions are centralized and distributed approaches, rather than hybrid schemes. As such, a detailed literature survey was carried out in relation to centralized and distributed approaches. In order to prove the efficiency of the proposed tree-based data aggregation, some of the clustering algorithms and opportunistic routing algorithms have also been studied and presented here.

2.1 CENTRALIZED APPROACH-BASED SCHEMES

In centralized approach-based schemes, the sink or base station collects the global information of the network topology and computes the optimal shortest path of the sensors before deploying them for data aggregation.

Mpitziopoulos, Gavalas, Konstantopoulos and Pantziou (2007) have proposed a near-optimal scheduling algorithm (NOSA) for identifying an appropriate number of mobile sensor nodes and their near optimal paths using the Esau-Williams heuristic. In NOSA, the parallel deployment of multiple agents is suggested, where each agent visits the subset of nodes. NOSA outperforms the single agent-based approaches (e.g., LCF, GCF and GA) in terms of data fusion cost and overall response time, but it experiences high computational complexity in determining the agents' itineraries.

In order to reduce high computational complexity, Chen Gonzalez, Zhang, and Leung (2009) has proposed a multi-agent scheduling (MAS) algorithm. To reduce the latency, the authors have proposed the MAS scheme because it helps in the collection of concurrent sensor data. These algorithms differ in cluster-group formation methods. In MAS, the authors used an angle gap for clustering all the sensor nodes in a particular direction as a single group. This approach does not describe how to determine the optimal angle gap threshold. Cai, Chen, Hara, and Shu (2010) and Chen, Cai, Gonzalez, and Leung (2010), has been proposed a genetic algorithm base mobile agents itinerary planning algorithm to form the clusters. Wang, Chen, Kwon, and Chao (2011) has proposed a genetic algorithm-based approach which has the limitation of its higher computational overhead. These algorithms assume that the set of sensor nodes to be aggregated by the sink are predetermined, which limits the application scope of the network.

Konstantopoulos, Mpitziopoulos, Gavalas, and Pantziou (2010) have proposed a greedy tree-based scheduling algorithm (TBSAs) to identify near-optimal paths for multiple agents. This algorithm is a centralized algorithm where the sink statically determines the number of aggregators and their schedules. The main theme of the TBSA algorithm is to divide the area around the sink into concentric zones, so that it can construct the near-optimal path tree from inner zones to outer zones. The main limitation of a centralized approach-based data aggregation scheme is that it uses static aggregation scheduling algorithms because it is based on an old view of the network topology. This approach lacks dynamic recovery from node or communication link failures.

2.2 DISTRIBUTED APPROACH-BASED SCHEMES

The distributed approach-based data aggregation protocol helps the aggregators to change the route dynamically according to the current state of the network.

Xu and Qi (2006, 2008) proposed a dynamic data aggregation algorithm for target-tracking applications. Energy consumption, information gain and the remaining energy of a node represent the cost function used in this method for selecting the next node. The cluster-head aggregates the data from its cluster, and then returns to the sink node. This algorithm is more time-consuming and may face difficulties in returning to the sink without additional forwarding information. In MADD, the authors Karaboga et. al (2012) and Chen, Kwon, Yuan, Choi, and Leung (2007) have proposed mobile agent-based directed diffusion, where an aggregator node visits a subset of nodes. In this model, the sink uses the first phase of the directed diffusion algorithm proposed by Intanagonwiwat, Govindan, and Estrin (2000) to determine the clusters. However, the actual data aggregation is carried out by dispatching an aggregator that sequentially visits the subset of nodes.

In software agent-based directed diffusion, node visits are determined at the sink node, although the authors

Shakshuki, Malik and Denko (2008) cannot access the procedure. This method uses the routing cost and the remaining energy of node in order to select the next node to be visited by an aggregator. The main limitation of the scheme is that, it depends on a directed diffusion scheme, which incurs extra communicational overheads for data aggregation and is only appropriate for request-based data aggregation applications. Gupta, Misra, and Garg (2012) has proposed a multiple mobile aggregator with dynamic scheduling-based data dissemination (MMADSDD) protocol, where nodes are organized in fixed regions and each aggregator is responsible for collecting aggregated data from each region. The route of an aggregator is dynamically located at each node using the cost function. MMADSDD adapts to unexpected node failures during data aggregation from the aggregator to the sink, although it consumes slightly more energy than TBID proposed by Konstantopoulos et al. (2010).

Boudia, Senouci, and Feham (2015) proposed an additive homomorphism encryption and an aggregate message authentication code (MAC) to provide end-to-end confidentiality and end-to-end integrity, respectively. SASPKC adopts state full public key encryption (SPKE) for efficiency in terms of computation and communication costs. SASPKC aggregates not only cipher texts but also signatures, while end-to-end data confidentiality and integrity security services are provided by using symmetric homomorphism encryption and an aggregate MAC, respectively. While considering new attacks, such as selective forwarding, SASPKC does not support node mobility. The main contribution of Sun, Luo, and Das (2012) is the proposed combination of a trust mechanism, data aggregation and fault tolerance to enhance data trustworthiness in wireless multimedia sensor networks, which consider both discrete and continuous data streams. Ho, Wright, and Das (2012) proposed a framework to detect compromised nodes in WSNs, as well as applied software attestation for the detected nodes. They reported that the revocation of detected compromised nodes cannot be performed due to the high risk of false positives in the proposed scheme.

Rezvani, Ignjatovic, Bertino, and Jha (2015) proposed a novel collusion attack scenario against the number of existing IF algorithms. The authors have proposed an improvement to the IF algorithms in terms of providing an initial approximation of trustworthiness of sensor nodes. When compromised aggregators are involved in data aggregation, this model fails in protecting the data. Moreover, this method is only suitable for the new deployment of data aggregation. In multi-channel scheduling algorithms, the authors Ghosh, Incel, Kumar, and Krishnamachari (2009) dedicated their efforts to the aggregation scheduling problem in WSNs when multiple frequency channels are available. The authors then demonstrated that finding the minimum number of channels required in the network to alleviate all interference is NP-hard. The NP hardness in minimizing the scheduling latency in an arbitrary network, with respect to multiple channels, has been proved. The work proposed by Joe, Choy, and Sheriff (2010) formulates the scheduling problem of minimizing the overall data transmission delay. The characteristics of optimal scheduling were studied, followed by the evidence from the lower bound of optimal performance. Two scheduling policies were proposed in the work. The decision in one policy was made based on the current system state, while the predication of future system conditions was also taken into consideration in the other policy.

Li, Guo, and Prasad (2010) offer a solution for the distributed aggregation scheduling problem in WSNs with respect to minimum latency. Compared with centralized solutions, a distributed scheduling plan has its own advantages. In this paper, an algorithm based on vertex coloring was proposed with a proven delay of $4R+2\Delta$ - 2.The minimum latency aggregation scheduling in WSNs with multiple sinks proposed by Yu and Li (2011) has been investigated. In contrast to what is found in the prior literature, this model proposed a dynamic selection of a sink node based on the shortest path by sensor nodes for the purpose of minimizing transmission latency. Two approximation algorithms with bounded latency were proposed. Zeydan, Kivanc, Comaniciu, and Tureli (2012) have proposed an adaptive and distributed routing algorithm for correlated data gathering, as well as exploiting the data correlation between nodes using a game theoretic framework. Routes are chosen in order to minimize the total energy expended by the network using the best response dynamics in relation to local data. The cost function used for the proposed routing algorithm takes into account the energy, interference and in-network data aggregation. This paper specifically addresses the problem of effective energy minimization, although the quantitative analysis of delay minimization has not been resolved.

Yao, Cao, and Vasilakos (2015) proposed an energy-efficient delay-aware lifetime-balancing protocol for data collection in WSNs. The authors proved that the problem formulated by EDAL is NP-hard, as well as proposed a centralized and distributed heuristic to reduce its computational complexity. Although EDAL achieves significant energy consumption and delay, this model only proposed to resolve energy issues in heterogeneous networks. Choa and Hsiao (2014) proposed a structure-free and energy-balanced (SFEB) data aggregation protocol, which consists of two phases. In first phase, the proposed model designates some nodes as aggregators in order to gather as many packets as possible. Then, these aggregators send the collected packets back to the sink in phase two. Sensor nodes that fail to send data to aggregators will also transmit their packets to the sink in phase two. This model requires location information of sensor nodes to avoid structure-based data aggregation. Location information can be obtained by applying a localization protocol. Alshahrany, Abbod, Alshahrani, and Alshahrani (2016) has proposed a web based data collection framework which integrates data from homogeneous and heterogeneous networks. But integrating WSN and RFID is a complex design process when supporting real-time applications. Sensor semantic network ontology has been used to form the cluster for data aggregation. Hence the cluster head should have good energy and computational capability. When implementing this model in remote and autonomous network applications, it leads to more energy wastage and complex process on cluster formation.

Karaboga et al. (2012) proposed a novel energy-efficient clustering mechanism based on an artificial bee colony algorithm in order to prolong the network lifetime. The artificial bee colony algorithm, which simulates the intelligent foraging behavior of honey bee swarms, has been successfully used in clustering techniques. The proposed ICWAQ protocol uses efficient and fast searching features of the ABC algorithm in order to optimize clustering of the nodes in the cluster-head selection process of defining routing gateways. Since this algorithm uses random cluster-head selection, it leads to collision problems. This algorithm is not suitable for routing mobile networks due to MAC layer issues.

2.3 OPPORTUNISTIC ROUTING

Opportunistic routing is widely known to have substantially better performance than unicast routing in wireless networks with lossy links. The opportunistic routing proposed by Mao, Tang, Xu, Li, and Ma (2011) allows any node that overhears the transmission to participate in forwarding the packet. The routing path is selected on the fly and completely opportunistic based on the current link quality situations. As stated by Chakchouk (2015), the existing opportunistic routing proposals can be categorized into three main classes: geographic, link state aware and probabilistic routing.

The geographic opportunistic routing proposals proposed by Wang, Chen, and Li (2012), Zeng, Yang, and Lou (2009) are location-centric. Hence, they are practical for scenarios where the knowledge of the nodes location is necessary, such as fire detection, gas leakage monitoring, and rescue operations. However, these protocols may not be efficient in terms of delay and reliability.

Link-state-aware opportunistic routing protocols has been proposed later by Rozner, Seshadri, Mehta, and Qiu (2009) and Han, Bhartia, Qiu, and Rozner (2011) to take other parameters, like link quality and bandwidth, into account. These protocols provide higher throughput for Wireless Mesh Networks and Internet applications. Sudarsono, Huda, Fahmi, Al-Rasyid, and Kristalina, (2016) has proposed an environmental health monitoring application through WSN with secure data communication access. For the communication module, they have used IEEE802.15.4-based communication which is legacy model for state of the art IoT applications. Moreover their proposed model is more specific to small scale applications where energy and security need not to be considered. When implement this model in large remote environment applications, the key sharing and renewal mechanisms which is proposed has leads to more energy consumption.

Conan, Leguay, and Friedman (2008) and Liu and Wu (2012) have proposed a Probabilistic opportunistic routing which is suitable for dynamic wireless networks, since it copes with the high mobility property of these networks by using online inference and prediction schemes to estimate the links' qualities and availabilities. But in more stable/stationary wireless networks, it has been observed that opportunistic routing design could be further refined and optimized using optimization-based formulations for candidate relays selection and prioritization.

However, with the proliferation of wireless heterogeneous networks and the security risks, a lot of research

work still needs to be developed in order to adapt opportunistic routing to these new challenging environments. This is particularly important, since some opportunistic routing applications, like disaster management, are prone to be deployed in similar environments.

To overcome all the drawbacks of the existing models, the authors Gopikrishnan and Priakanth (2016) proposed a unique solution namely, hybrid secure data aggregation (HSDA), which can resolve security and energy issues because it provides highly secure data aggregation in an energy-efficient way. The HSDA implements an end-to-end symmetric key cryptography for secure authentication by using a shared public key, as well as hop-by-hop asymmetric key cryptography with the private keys of each node for data integrity and confidentiality. Although the authors have successfully resolved the maximum issues in HSDA, multiple algorithms implementation renders HSDA algorithm as inefficient in terms of energy utilization.

2.4 MOTIVATION

With regard to WSNs, data aggregation routing protocols have always been a major research topic, although the unique solutions that resolve maximum issues, such as collision, delay, energy and security factors, are not addressed by existing protocols. The literature survey conducted in relation to this article analyzed all the data aggregation issues, with the best solutions presented for the existing models. EECD Zeydan et al. (2012) presents an effective solution, which resolves energy and collision problems. When the network density is greater, EECD fails in reducing the aggregation delay and produces less significant throughput. A centralized and distributed algorithm, which resolves energy and delay issues, has been proposed in EDAL Yao et al. (2015), although the authors specifically presented their protocol only for heterogeneous networks. To avoid structure-based data aggregation, the authors have presented a modified aggregator-based data aggregation known as SFEB Chao and Hsiao (2014). Although energy-efficient data aggregation has been achieved in SFEB, it fails with regard to fast data aggregation due to the mediatory implementation of the localization algorithm. In ABC, the authors Karaboga et al. (2012) have presented a cluster-based data aggregation algorithm, which mainly concentrates on reducing aggregation delay. The randomized cluster-head selection in ABC results in insecure data aggregation. To overcome all the drawbacks of existing models, as well as to resolve the maximum issues in order to obtain a unique solution, Gopikrishnan and Priakanth (2016) have proposed HSDA. Although the HSDA resolves all data aggregation issues, the energy utilization of this algorithm motivates to further reduce the energy utilization by reducing the communication and computational overheads of sensor nodes.

3. SYSTEM MODEL

This section briefly describes the network model and energy model assumed in the proposed algorithm.

3.1 THE NETWORK MODEL

We represent a WSN by a connected graph $G\ (V, E)$, where the vertex set V corresponds to the nodes in the network, while the edge set E corresponds to the wireless links between nodes (we use "edge" and "link" interchangeably hereafter). Let n and l be the cardinalities of V and E, respectively. There is a special node $s \in V$, which represents the sink that collects data from the whole network. To simplify the network model, a few reasonable assumptions has adopted as follows:

1. There are N sensor nodes that are distributed in the M × M square field.
2. All the nodes and the BS are stationary after deployment.
3. All the sensor nodes can be heterogeneous, although their energy cannot be recharged.
4. All the sensor nodes are location-unaware.
5. All the nodes can use power control to vary the amount of transmitting power.
6. The BS is out of the sensor field. It has a sufficient energy resource, while its location is known by each node.
7. Each node has an identity (ID).
8. The WSN is considered to be a data aggregation tree, which is a connected a cyclic graph G (V, E).

3.2 ENERGY MODEL

The energy model used in this paper is similar to the energy model used in Chen et al. (2009). The energy that is consumed in order to transmit k-bit data from S_i to S_j over a distance d is given by,

$$E_{tx}(S_i, S_j) = \begin{cases} \left(E_{elec} + \in_{fs} \times d^2\right) \times kford < d_0 \\ \left(E_{elec} + \in_{mp} \times d^2\right) \times kford \geq d_0 \end{cases}$$

where E_{elec} is the energy dissipated in communicating 1- bit data, while $\in_{fs}$, $\in_{mp}$ are energies necessary for the amplifier in the free space and the multipath model, respectively. The energy required for the reception of k-bit data by a sensor node S_j is given by,

$$E_{Rx}(S_j) = E_{elec} \times k$$

4. HYBRID COMMUNICATION FOR DATA AGGREGATION

The hybrid communication for the energy-efficient data aggregation algorithm consists of three phases: i) energy-efficient aggregation tree construction, ii) a collision-free data aggregation algorithm, and iii) an encryption and decryption algorithm for secure data aggregation. Table 1 represents the terminologies used in the proposed algorithms.

4.1 HCED PHASE-1: AGGREGATION TREE CONSTRUCTION

In the HCED approach, the tree construction follows a binary search tree construction method. In this phase, each node begins the tree construction process when a node receives the construction request message Con_Req (P_N, P_{LN}, P_{RN}) from the parent node. After receiving a construction request from other nodes, each node broadcasts a Node_Msg with the following two values: one is the node id, and the other is the residual energy of this node within radio range r. At the same time, each node receives the Node_Msg messages from its neighbor nodes. Based on the Node_Msg broadcasting, the neighbor nodes will be discovered along with their residual energy by each node N_i. The response time of each neighbor node in the sensing limit of each node will be registered in the routing table of node N_i. If node υ has the sensing limit of d meters, υ will identify all the neighbors$N_\upsilon = \{V\} \in G$existing within d meters. Similarly, all the nodes present in the network will identify their neighbors and their distances during the broadcast.

After the neighbor discovering process, the data aggregation tree construction begins at each node. As stated earlier, this HCED approach implements a binary search tree-based algorithm for constructing an energy-efficient data aggregation tree. Initially, the construction request begins from sink node S. The sink node is the root node that is responsible for aggregating data from other nodes with high energy resources. The root node first identifies the left child, which has the shortest response time among all the neighbor nodes and in turn identifies its right child, which has the second shortest response time in the routing table. If the node with the least response time is already rooted with other parent nodes, the second least responsive node will be assigned as the left child, while the third least responsive node will be assigned as the right child.

In a binary search tree, a node can have a minimum of zero and a maximum of two child nodes. Now that the root node has attained its maximum child limits, the construction control can be moved to the left child of the root node through a Con_Req (P_N, P_{LN}, P_{RN}) message. The left child node then follows the same binary search tree construction algorithm until it finalizes its left and right child nodes. In turn, the construction control is moved to the right child of the root node and performs the same procedure until it finalizes its right and left child nodes. This process will continue until all the nodes finalize their child nodes. The construction control subsequently moves to the next node or the next level only when a node finalizes its left and right child node or when a node has only one left child. When a node finds its child nodes and their parent nodes, the construction control is moved to the left child. If all the neighbors of a node are associated with another parent node, that node is assigned as a leaf node. When all the nodes identify their child and parent nodes, the tree construction process can be stopped.

Algorithm 1 has been developed as per the description of the tree construction phase. This algorithm must be implemented for all sensor nodes that are deployed within the network, after which the algorithm execution begins at the root node with a Node_Msg broadcast to the network. In Algorithm 1, the following assumptions are made:

Table 1. Terminologies used in algorithms.

Terminologies	Description
$N_\upsilon = \{N_1, N_2 \ldots N_n\} \in G$	The list of N neighbour nodes
C_N	The current node where the tree construction process
Con_Req(P_N, P_{LN}, P_{RN})	Tree construction request message.
P_N	The node where the current node receives the tree construction request.

Table 1. Terminologies used in algorithms. (cont…)

Terminologies	Description
P_{LN}	The left child of the parent node P_N.
P_{RN}	The right child of the parent node P_N.
Root	The data aggregation path defined by the tree construction algorithm
L_N	Left Node of current node C_N
R_N	Right Node of current node C_N
N_N	The node where the tree construction request has to be send from P_N.

4.2 HCED PHASE-2: COLLISION FREE DATA AGGREGATION ALGORITHM

After the successful construction of the energy-efficient data aggregation tree, the root node is responsible for initiating the data aggregation process among the network. The initiation of data aggregation process will vary based on the type of application. If the application is event-driven, the node that identifies the event within its sensing area will begin the data-sending process towards the root node through their root nodes. In a time-driven application, the root node initiates the data aggregation process based on the time interval.

Case1. If the application is event-driven; the data aggregation process will only follow the routing table when sending the data towards the root node. In this case, a node N_i, which has sensed the data, follows a three-step procedure:

- Step1 –First, encrypt the data using Algorithm 5.
- Step2 –Based on the routing table, the encrypted data are simply forward to their parent node P_N.
- Step3 –Furthermore, all the intermediated nodes follow the aggregation tree to send the encrypted data to the root node.

Here the proposed energy-efficient data aggregation tree is enough to aggregate the data in an efficient way.

*Case2.*If the application performs the data aggregation based on the time interval, the data aggregation process will be initiated by the root node. In this case, before the data aggregation process initiated at the root node, it should ensure that:

- All the nodes have finalized their parent PN and child nodes LN or RN
- All the nodes have the routing table constructed by Algorithm 1.

When all the nodes have the required information, the root node will initiate the data aggregation process by broadcasting its public key Q_k among the network. When a node receives the public key Q_k, the following applies:

- Step1 –The node Ni will execute the data aggregation algorithm (Algorithm 2) in order to collect the data from its child nodes.
- Step2 –After aggregating the data from the child nodes, the node Ni generates its own data to send.
- Step3 –If the node Ni does not have its own data to send, it simply forwards the encrypted data, which are received from its child nodes.
- Step4 –If the node Ni has its own data to send, it executes an encryption algorithm (Algorithm 5) to encrypt its own data with the data aggregated from the child nodes.
- Step5 –After encrypting the data, the node Ni forwards the encrypted data to its parent node.

Similarly, all the nodes perform the same procedure in order to collect and send the data to the root node.

Algorithm1: Hybrid data aggregation tree construction algorithm

Ensure:

- Sensor nodes are deployed randomly with Algorithm 1, 2 and 4.
- All the sensor nodes are registered with Sink node through a Node ID
- All the senor nodes have same transmission range and initial energy.

Output:

- Data Aggregation Tree
- Routing Table for each node N_i

```
1.   Begin
2.   Con_Req (PN, PLN, PRN)
3.
4.   If (CN =PLN)
5.   CN – Root ← PN-Left
6.   Else if (CN = PRN)
7.   CN – Root ← PN-Right
8.
9.   Broadcasting by CN to find Nυ
10.
11.  i ← 1 to N in Nυ
12.
13.  While (Ni-Root is NA) then
14.  {
```

15. If ($N_i = P_{LN}$ || $N_i = P_{RN}$ || $N_i = P_N$)
16. {
17. i ← i+1
18. }
19. Else
20. {
21. $L_N \leftarrow N_i$
22. N_i-Root ← C_N-Left
23. i ← i+1
24. While (N_i-Root is NA) then
25. {
26. If ($N_i = P_{LN}$ || $N_i = P_{RN}$ || $N_i = P_N$)
27. {
28. i ← i+1
29. }
30. Else
31. {
32. $R_N \leftarrow N_i$
33. N_i-Root ← C_N-R_i
34. }
35. }
36. }
37. End If
38. }
39. If ($C_N = P_{LN}$)
40. $N_N \leftarrow P_{RN}$
41. Else
42. $N_N \leftarrow L_N$
43. Send Con_Req (P_N, P_{LN}, P_{RN}) to N_N
44. End

In this proposed system, the collision-free data aggregation algorithm will be executed by all the nodes in the network. Initially, this algorithm requires the routing table (generated by Algorithm 1), encrypted data and the list of nodes U_A={N_{n1}, N_{n2}...N_{ni}} to be aggregated in the network. The time slot T_S is the time required by the sink node to aggregate data from all the nodes; T_S will be initiated to zero. First, the algorithm verifies whether the node is a leaf node or not. If the node Ni is a leaf node, as well as a left node for another parent node, it will send the data to its parent node during the first time slot. Then, the node Ni will be removed from the set UA. In turn, the node, which is a leaf node, as well as a right node for another parent node, will send the data to its parent node during the second time slot. This node will also be removed from the UA. This aggregation process will be continued until all the nodes are removed from the UA. Once all the data are aggregated by the sink node, the value of T_S is the time required by the algorithm to aggregate the data from all the nodes.

Algorithm 2: Secure data aggregation algorithm

Define
- Routing table to be the details of node representing Node-ID, Parent-ID, Left Child-ID and Right Child-ID
- U_A = {N_{n1}, N_{n2}....N_{ni}} is to be set of nodes to be aggregated in the network.
- R_T = {R_1,R_2...R_n} is the list of neighbour nodes' root of a Node Ni
- T_S is to be the Time Slot required to complete the aggregation process
- CT_N is the encrypted cipher data of the actual data

Require:
- Routing Table
- U_A = {N_{n1},N_{n2}....N_{ni}}
- Encrypted cipher data CT_N.

Initialize:
- T_S=0
- C_N = N_{ni}in R_T
- Flag = 0
- Counter = $R_{T\text{-count}}$

1. Begin
2. Do until U_A-count > 1
3. {
4. Do until Counter >= 1
5. {
6. If (N_i-Root = C_N-Left)
7. {
8. If (N_i-Root =C_N-Right)
9. {
10. Flag=1;
11. }
12. }
13. If (Flag=0)
14. {
15. C_N sends data to P_N
16. Remove C_N from U_A
17. U_A-count - -
18. }
19. Remove C_N-id from R_T
20. Counter - -
21. }
22. T_S=T_S+1
23. Counter = $R_{T\text{-count}}$
24. If (Flag=1)
25. {
26. C_N sends data to P_N
27. Remove C_N from U_A
28. U_A-count - -
29. Stop
30. }
31. }
32. End

Algorithm 3: Public key generation algorithm

Requires:
- Initial hours (T_h) and minutes (T_m) of data aggregation process.

Output:
- public key (Q_k) of root node

```
1.  Begin
2.      For Qk1 = 2 to Th
3.      {
4.          If (Th mod Qk1 = = 0)
5.          {
6.                  Th = Th / Qk1
7.                  Qk1=Qk1-1
8.          }
9.      }
10.
11.     For Qk2 = 2 to Tm
12.     {
13.         If(Tm mod Qk2 == 0)
14.         {
15.                 Tm = Tm / Qk2
16.                 Qk2=Qk2-1
17.         }
18.     }
19.     Qk = Qk1 + Qk2
20. End
```

4.3 *HCED PHASE-3: ENCRYPTION AND DECRYPTION ALGORITHM*

This third phase of the proposed HCED algorithm describes the encryption and decryption algorithm used to provide secure data aggregation. After completion of the data aggregation tree construction process by Algorithm 1, the public key generation algorithm (Algorithm 3) will be executed by the root node as an initial process of data aggregation. In Algorithm 3, the initial hour (T_h) and minutes (T_m) of the data aggregation process are the key factors. By performing the mathematical calculation on T_h and T_m, Algorithm 3 finds the public key Q_k for a single-time data aggregation process. Whenever the data aggregation process is initiated by the root node, it generates a new public key Q_k and broadcasts it among the network through a data aggregation request.

The encryption algorithm (Algorithm 4) used in this proposed model will be executed by all the nodes in the network. If the node is a leaf node, it will first execute the encryption algorithm in order to encrypt the plain data PT_{Ni} and the MAC M_{Ni}. The MAC is the authorized code to ensure the integrity of the message for each node. The MAC is appended with the actual data before it is sent to other nodes. To reduce the communicational overhead among the network, this encryption algorithm only requires the public key Q_k of the root node. In order to provide a high level of confidentiality with the data, each node generates its own private key P_k based on the node ID and the public key Q_k. The proposed method does not have any separate algorithm for generating the private key to reduce the computational overhead on the nodes. Instead, it generates the private key, which then encrypts the plain data and the MAC by using Q_k and P_k in a single encryption algorithm.

Algorithm 4: Encryption Algorithm

Requires:
- Plain data (PT_{Ni}) of the sensor node N_i
- Public key shared by root node Q_k
- Message authentication code (M_{Ni}) of node N_i

Output:
- Encrypted cipher data (CT_{Ni}) of each node N_i
- Encrypted message authenticated code (EM_{Ni}) of each node N_i

```
1.  Begin
2.      For i=2 to Qk
3.          {
4.              Temp3=Temp1+Temp2;
5.              Temp1=Temp2;
6.              Temp2=Temp3;
7.           }
8.      For Pk1 = 2 to Temp3
9.          {
10.         If(Temp3 mod Pk1 == 0)
11.         {
12.        Temp3 = Temp3 / Pk1
13.        Pk1=Pk1-1
14. }
15.         }
16.     Pk← (Pk1 x Node-Id) Mod Qk
17.     EMNi+ ← MNi x (Temp3 ModPk)
18.     CTNi ← (PTNi x Pk) + EMNi
19. End
```

After aggregating the data from all the nodes, the root node will execute the decryption algorithm (Algorithm 5) to find the actual data. Before decrypting all the data, the root node must have the encrypted cipher data (CT_{Ni}) and encrypted MAC (EM_{Ni}) for each node N_i. Since the public key is generated by the root node, it has the information about Q_k, but it does not have the information about P_k. The key idea in this decryption algorithm is to find the private key P_k using the node ID by following the same

procedure. As such, Algorithm 5 first identifies the private key P_k for each node N_i, then it decrypts the encrypted data CT_{Ni} and encrypted MAC EM_{Ni}.

Algorithm 5: Decryption Algorithm

Requires:
- Encrypted cipher data (CT_{Ni}) of each node N_i
- Encrypted message authenticated code (EM_{Ni}) of each node N_i
- Public key Q_k which is generated by root

Output:
- Decrypted cipher data (PT_{Ni}) of each node N_i
- Decrypted message authenticated code (M_{Ni}) of each node N_i

```
1.  Begin
2.      For i=2 to Qk
3.          {
4.              Temp3=Temp1+Temp2;
5.              Temp1=Temp2;
6.              Temp2=Temp3;
7.          }
8.      For Pk1 = 2 to Temp3
9.          {
10.             If(Temp3 mod Pk1 == 0)
11.             {
12.             Temp3 = Temp3 / Pk1
13.             Pk1=Pk1-1
14.             }
15.         }
16.     Pk← (Pk1 x Node-Id) Mod Qk
17. MNi← EMNi / (Pk1 mod Pk)
18. PTNi ← (CTNi – EMNi) / Pk
19. End
```

5. HCED ALGORITHM DESCRIPTION

The theoretical description for the proposed model has been presented in this section with a sample network. In the process, a forest fire detection system has been considered, which is implemented by using WSNs to prevent forest fire hazards. In forest fire detection systems using a WSN, sensor nodes aggregate measurement data, such as relative humidity, temperature, smoke and wind speed, in order to determine the forest fire danger rate. One of the major functions of a forest fire detection system is to gather the information about the forest fire as early as possible. When implementing a WSN for forest fire detection, perfect data aggregation, which resolves problems such as massive energy waste, collision, aggregation delay and localization of the nodes, is required.

To describe the proposed model, consider a small portion of the application. Figure 1 show a sample network with 15 nodes along with one sink node, which is also deployed within the network. Assume that all the nodes are randomly deployed on the monitoring area, with each sensor node having an equal sensing limit. A node that is connected to a gateway and has a high energy resource will be assigned as the sink node. Since all the nodes are randomly deployed, they are unaware of their neighbors and their locations. As per the key idea in the proposed model, each sensor node is deployed with Algorithms 1, 2 and 4 in it. Before deploying the sensor nodes, each sensor node is registered with the sink node via a node ID. Since the sink node does not need to send any data to any other parent nodes, it does not require any encryption algorithm. Therefore, the sink node will be deployed with Algorithms 1, 2, 3 and 5.

Initially, Algorithm 1 is executed by the sink node S1 with Con_Req (P_N, P_{LN}, P_{RN}). Since the current node C_N is the sink node, the condition from lines 4 to 8 will not be true. As such, the Node_Msg will be broadcast at line 9. The neighbor nodes will be identified based on the responses to the Node_Msg request. Based on the response time of the neighbor nodes, Algorithm 1 arranges the neighbor nodes in a sequence. The sink S_1 identifies $N_\upsilon = \{N_3, N_2, N_1\}$ as its neighbor set. Lines 15 to 18 identify whether the neighbor nodes are associated with other parents or not. If the first neighbor node does not have a parent node, lines 21 to 23 identify the left child, while lines 26 to 36 identify the right child. Here, S_1 identifies N_3 and N_2 as the left and the right child, respectively. Then, S_1 sends the Con_Req (P_N, P_{LN}, P_{RN}) to the left root N_3. Table 2 represents the routing table for the sink S_1.

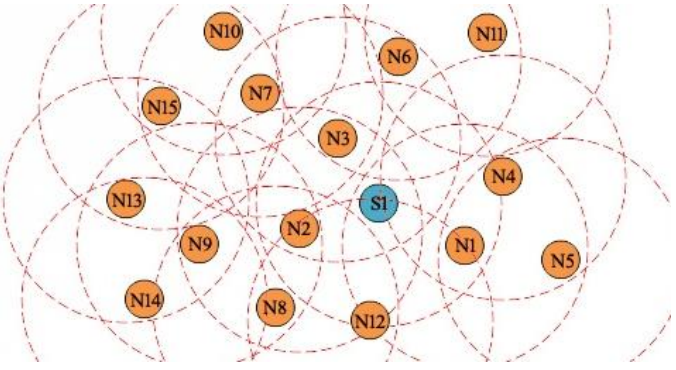


Fig. 1. A sample network model in Forest Fire detection application.

Table 2. Routing Table of Sink node S1.

Node-ID	Parent-ID	Neighbor Nodes	Root (R_T)
S_1	NA	N_3	S_1_Left
		N_2	S_1_Right
		N_1	NA

Likewise, the data aggregation tree has been constructed based on the Con_Req (P_N, P_{LN}, P_{RN}) message. Figure 2 represents the data aggregation tree constructed by Algorithm 1 for the sample network considered in Figure 1. Tables 3 and 4 represents the routing table for nodes N_3 and N_2, respectively.

Table 3. Routing Table of Sink node N3.

Node-ID	Parent-ID	Neighbor Nodes	Root (R_T)
N_3	S_1	S_1	Parent
		N_7	N_3_Left
		N_2	S_1_Right
		N_6	N_3_Right

Table 4. Routing Table of Sink node N2.

Node-ID	Parent-ID	Neighbor Nodes	Root (R_T)
N_2	S_1	S_1	Parent
		N_8	N_2_Left
		N_3	S_1_Left
		N_9	N_2_Right

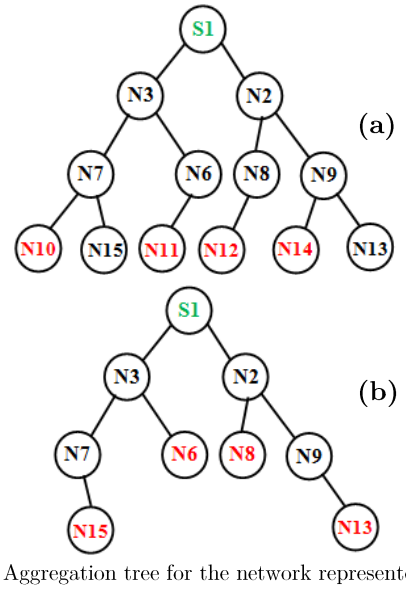


Fig. 2. Aggregation tree for the network represented in Figure. 1. (a) Data Aggregation in 1st time slot. (b) Data Aggregation in 2nd time slot.

After constructing the data aggregation tree, the sink S_1 is responsible for initiating the data aggregation process. S_1 will initiate the data aggregation process by sending the public key to its child nodes. As such, Algorithm 3 will be executed by S_1 to generate the public key Q_k. Consider the initial time of data aggregation to be T_h=23 and T_m=55. Lines 2 to 9 in Algorithm 3 will generate the public key Q_{k1}=23, while lines 11 to 18 will generate Q_{k2}=11. Then, line 19 will generate the public key Q_k=34.

Now S_1 can initiate the data aggregation process by broadcasting Q_k among the network. When a node N_i receives the public key, it starts executing Algorithm 4 in order to encrypt the plain data PT_{Ni}. When considering the node N_7, the plain data are PT_{N7}=14 and the MAC is 7. The received public key Q_k=34. Based on the Q_k, lines 8 to 15 identify P_{k1}. Based on P_{k1}, line 16 identifies the private key P_k=14. Line 17 encrypts M_{N7} to find EM_{N7}=56, while line 18 encrypts the plain data PT_{N7} to find the cipher data CT_{N7}=588. Now, the node N_i executes Algorithm 2 for sending data towards the sink. The routing table of each node will be the key factor in Algorithm 2. Based on the routing table, line 6 and 8 check whether the node is to the right or left of its parent node. If the node is left and a leaf node, lines 13 to 19 will send the data to its parent in T_S=1. If the node is a leaf and its parent is free to receive the data, lines 24 to 30 send the data to its parent in T_S=2 and stops Algorithm 2. In Figure 2(a), the nodes N_{10}, N_{11}, N_{12} and N_{13} will send the data in the first time slot and remove all the nodes in Figure 2(b). In the second time slot, the nodes N_6, N_8, N_{13} and N_{15} send the data in the second time slot. The sample network considered in Figure 1 will aggregate all the data in T_S=5.

When the sink node has aggregated all the data from the network, it begins the decryption process by executing Algorithm 5. Each node's ID is the key factor in the decryption algorithm. Since the sink node has the public key Q_k, it needs to find the private key P_k. Lines 2 to 16 in Algorithm 5 find the private key P_{ki} of each node N_i. With the help of Q_k and P_{ki}, lines 17 and 18 decrypt EM_{Ni} and CT_{Ni} to find M_{Ni} and PT_{Ni} of the node N_i.

5.1 MAINTAINING THE AGGREGATION TREE

The data aggregation tree scheduling can be re-constructed periodically by all the nodes. This paper proposed a detailed algorithm that can be employed on

every node. So that, each node can easily re-construct the energy efficient data aggregation tree. When energy level of nodes is below data transmission threshold, an active sensor periodically broadcasts its energy level to its neighbors. Based on the energy level each node can be re-constructing the tree for better performance.

In case of node failure due to energy or any other issues, the node cannot do broadcasting. Whenever a parent node is not receiving any broadcasting reply or data from its left or right child, the parent node will broadcast a request message to its entire neighbors for re-constructing the aggregation tree based on the available active neighbor nodes. And the information about the failure node will be appended to the request message. So that, node failure can be managed by the root or sink node.

6. MATHEMATICAL ANALYSIS

6.1 THEORETICAL ANALYSIS

As discussed in relation to the proposed algorithm in Section 5, the proposed HCED approach is constructing a binary tree-based energy-efficient data aggregation tree. Hence, it is required to prove that the proposed data aggregation has constructed a binary tree. Let the proposed data aggregation algorithm construct a G tree.

Theorem 1. If G is a tree, then every two nodes are joined in a unique path

Proof: Consider P_1 and P_2 are the two paths between the nodes N_1 and N_2.Let x and y be the first and second points on both paths when tracing the two points from node N1 to node N2. The paths between x and y on P_1 and P_2then make a cycle. But this cannot happen if G is acyclic.

Now it is mandatory to prove G is acyclic. A G tree is acyclic only when the tree is connected.

Theorem 2. G is connected

Statement: If every two nodes of G are joined by a unique path, then G is connected.

Induction: Let $n=e+1$.

Proof: Assume it is true for less than n points. Removing any edge from G breaks G into two components, since the paths are unique. (Theorem 1) when the sizes are n_1 and n_2, such that $n_1+n_2=n$. By using the induction hypothesis, $n_1=e_1+1$ and $n_2=e_2+1$. However:

$n = n_1+n_2$becomes

$n = (e_1+1)+(e_2+1)$

$= (e_1+e_2)+2$

$= e- 1+2$

$= e+1$

Hence, G is connected.

Theorem 3. G is acyclic

Statement: If G is connected, then G is acyclic.

Proof by contradiction: Suppose G has a cycle of length k. Then, there are k points and k edges on this cycle. Since G is connected for each node v that is not on the cycle, this results in the shortest path from v to a node on the cycle. Each such path contains an edge e_v that is not found on any other. Thus, the number of edges is at least e>= (n-k)+k=n, which contradicts the assumption n=e+1.

Corollary 1.

A graph G with n vertices is a tree T if any of the following conditions is satisfied:

- A tree is an undirected, connected and acyclic graph.
- The graph is connected and has exactly $n-1$ nodes.
- The graph is maximal without cycles.
- If so, Theorems 1, 2 and 3 proved all the three points. Hence the initial assumption is true.

Theorem 4. There is at least one grandparent node between any two parent nodes in the HCED aggregation tree

Proof: Consider a node P_N, which becomes a parent during the role assignment process. Without any loss of generality, suppose that P is included in the neighbor list of grandparent $U = \{N_{n1},N_{n2}...N_{nn}\} \in G\ (V,E)$. According to Algorithm 1, P changes its role to a child if P is the least or the second least node in the sorted neighbor list $U_s=\{N_1,N_2...N_n\}$. However, the instructions in Algorithm2 can only remove children and grandchildren, as opposed to the parent node (like P_N), from U_s. As such, P_N is rolled as a left child or right child in the neighbor list of grandparents, meaning that there is at least one grandparent node between any two parent nodes in the proposed aggregation tree.

Theorem 5. Under the two-hop interference model, the sink can receive all the aggregated data in most $O(R+\Delta)$ time slots

Note that under the two-hop interference model, any two x and y senders are unable to be a communication neighbor (otherwise, x will cause interference in the receiver of y).As such, given Δ neighbors of a node, need at least Δ / five time slots to simply let each of these Δ neighbors transmit once. Thus, the following theorem applies.

Theorem 6. Under the two-hop interference model, for any data aggregation method, it will take at least max(R; $\Delta/5$) time slots for the sink to receive the aggregated data

For the k-hop interference model, where k ≥ 3, then any two x and y nodes that are neighbors of a node u clearly cannot transmit simultaneously. Thus, Δ is a lower bound on the delay in data aggregation.

6.2 PATH COST ANALYSIS

The maximum path cost from any leaf node to a parent node is D_i.

$$D_i = max\{d(L_{Nj}, P_{Nj})\}\forall j$$

The maximum path cost from any leaf node to sink is D_{sink}.

$$D_{sink} = max\{d(L_{Nj}, N_{sink})\}\forall i, j$$

The initial staggered time out is calculated as follows: if $T_i$is the stagger timeout for the aggregator node i, then

$$T_i = T_{ci} + T_{Si}$$

where T_{ci} is the cascading timeout, which depends on the level of the parent node. This gives the initial timeout, as in the cascade timeout, for each node, which is the same for all the nodes in the same level. The $T_{Si}$is the aggregation timeout of the node, which depends on the number of children it has.

$$T_{ci} = 2 \times [T - (T_{TD} \times h)]$$
$$T_{Si} = (P_i/P_{sink}) \times (T - T_{TD} \times \Delta)$$

Here, h denotes the hop distance of the node A_{Nj}, Δ is the depth of the tree, T is the data generation period or the dead line and T_{TD} is the one-hop delay between the levels. It depends on the queuing delay, MAC delay, processing delay for aggregation function and the transmission delay.

It is assumed to be 0.1 seconds as used in existing model Yu and Li (2011). After introducing this initial delay in the aggregation timer, the aggregation timer is fired for every T seconds, thus enabling the collection of packets generated in that round from all the nodes in the collection tree.

6.3 PACKET LOSS ANALYSIS

The data loss analysis of the proposed system has been analyzed in the part dealing with the results analysis. The throughput of the sensor network using the HCED approach is evaluated in terms of the number of successful packet deliveries. It is directly related to packet loss. The packet loss ratio of the network can be calculated as follows:

$$\text{Packet loss ratio} = \frac{\text{Total data sent} - \text{total data received}}{\text{Total data sent}}$$

Finally, the traffic overhead of the proposed protocol is also analyzed in terms of communication and computational overhead. To evaluate the traffic overhead of the distributed approach in a WSN, the average amount of traffic transmitted within the network is tested.

6.4 COMPLEXITY ANALYSIS

If the aggregation tree has an n-node and one root, the left sub tree will have an i-node, while the right sub tree has an (n-i-1) node as per the proposed tree construction. Hence, the complexity of the proposed data aggregation will be defined as:

$$T(n) = (n-1) + T(i) + T(n-i-1)$$

The root contributes 1 to the path length of each of the order n-1 nodes. Therefore, the average overall complexity for n-nodes will be:

$$T(n) = (n-1) + \frac{2}{n}(T(0) + T(1) + \cdots + T(n-1)$$

where$T(n)$is $O(n \log n)$. Therefore, the average complexity of data aggregation over the network which has n–nodes is:

$$T(n)/n = O(n \log n)$$

7. RESULT AND ANALYSIS

In this section, the performance evaluations of the proposed model have been presented. The simulation results were compared and analyzed with existing models by simulating forest fire detection application scenario. The simulations are carried out using MATLAB on a personal computer with Intel Core i3 6th Gen processor and 4GB of RAM. The simulation parameters are given in Table 5. The initial node deployment in the designated area is implemented according to the parameters given in Table 5.

By varying the number of nodes from 100 to 600, the simulation scenarios have been changed to evaluate the performance of the proposed model in all the parameters, such as collision, delay, throughput and energy. For all simulation scenarios, only one node was selected as a sink node, which has high residual energy and is attached to a gateway. The sink is located anywhere in surveillance area. All the nodes in the network generate a broadcast reply with the size of 64 bytes. Each source generates random data reports with the size of 136 bytes when it senses data with a constant bit rate of one packet per second.

Based on the simulation results, the performance evaluation has been undertaken by comparison with the following existing models: EDAL Yao et al. (2015), ABC Karaboga et al. (2012), SFEB Chao and Hsiao (2014), EECD Zeydan et al. (2012) and HSDA Gopikrishnan and Priakanth (2016).

Table 5. Simulation Parameters.

Parameters	Value
Monitoring the field size	300 × 300
Number of nodes	100 to 700 in increment of 100
Node density	0.0052 nodes/m2
Transmission range	25 m
Simulation time	30 min
Initial battery power	25J
the transmit power	0.66W
the receive power	0.39W
the idle power	0.035W
Number of data packets send by each node	100 to 300 in increment of 100
Data size generated by each node	136 bytes
MAC protocol	TMAC

7.1 COMMUNICATIONAL OVERHEAD

The communicational overhead of a network describes the collision problem in the network. The HCED approach is a unique solution that resolves maximum issues including collision. Here, the simulation has been carried out in order to evaluate the collision in terms of energy. The simulation model initially consists of 100 nodes, while the simulation is continued until up to 100 packets are received. Figure 3(a) represents the additional average energy required by the sensor nodes when aggregating the 100 packets. The HCED approach saves a minimum of 3.5% more energy than other models regarding the communicational overhead when the number of nodes is 100. It also saves a minimum of 7.3% more energy when the number of nodes is 600. Figure 3(b) represents the additional average energy required by the sensor nodes when aggregating the 600 packets. The two-hop tree-based data aggregation resolves the collision problem, meaning that the proposed model reduces the communicational overhead.

7.2 COMPUTATIONAL OVERHEAD

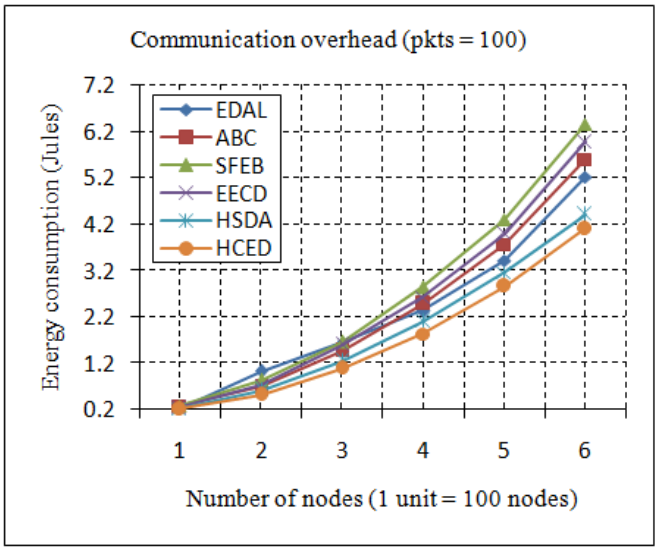
The additional energy required by a sensor node to execute the algorithms in a data aggregation model can be represented as computational energy. This computational overhead will be analyzed in relation to the proposed model in two cases. Case 1 will be evaluated until up to 100 packets are received from 100 to 600 nodes. The computational overhead comparison for the Case 1 will be described in Figure 4(a). Case 2 will be evaluated until up to 200 packets are received from 100 to 600 nodes. Figure 4(b) describes the computational overhead of the proposed model. When 100 packets are aggregating from 100 nodes, the proposed model reduces the computational overhead between 11% and 90% more than other models. When receiving the 200 packets, the HCED approach will save between 14% and 29% more energy than other models. When compared to all the other models, the proposed model will have less computational energy due to the simplified algorithms used for data aggregation and tree construction.

7.3 PERFORMANCE ON ENERGY CONSUMPTION

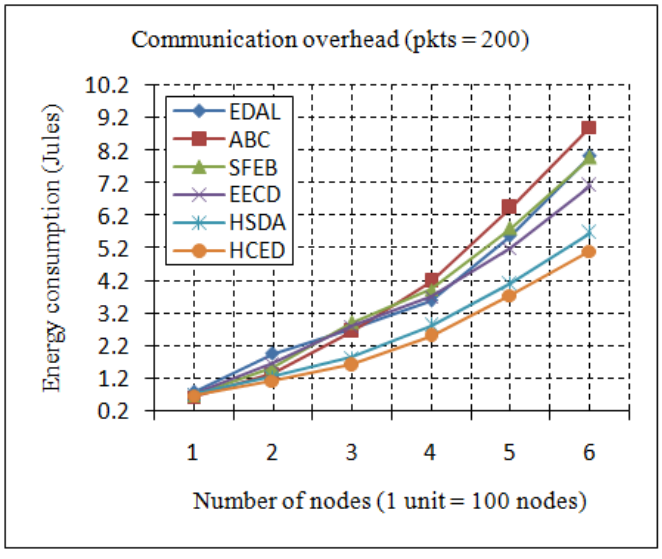
Two simulation scenarios have been considered in order to evaluate HCED performance regarding energy. In the first scenario, the number of nodes varies between 100 and 600 in increments of 100 nodes. The simulation was continued until 100 packets of data were received, after which the average energy consumption was evaluated to aggregate 100 packets. Based on the energy consumption, the results were compared with other models and are described in Figure 5(a). From the simulation results, it can be seen that the collision-free binary tree-based data aggregation model in the HCED approach reduces a minimum of 2.4% less energy consumption than the HSDA and a maximum of 36% less energy consumption than the ABC model when the number of nodes is 100. When increasing the number of nodes up to 600, the resultant energy consumption of the HCED approach is 11% less than the HSDA and 104% less than the ABC model.

In the second scenario, the simulation study was conducted until 200 packets were received, after which the average energy consumption was evaluated. The simulation results in this scenario have been studied and are described in Figure 5(b). When the number of packets is increased, the proposed model saves 6% more energy

than the HSDA model and 38% more than the ABC model when there are 100 nodes involved. When the number of nodes is 600 and the number of packets is 200, the HCED approach saves 6.9% more energy than the HSDA model and 91% more than the ABC model. This comparison study on energy consumption with other models proves that the proposed model improves energy utilization because of its collision-free binary tree-based tree construction and simplified data aggregation algorithm.

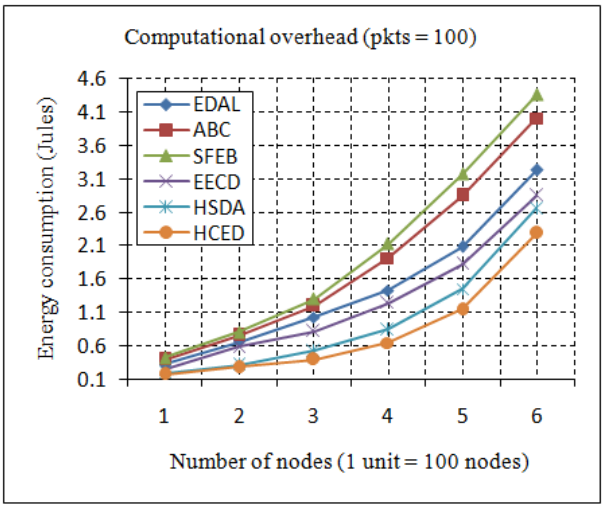


(a)

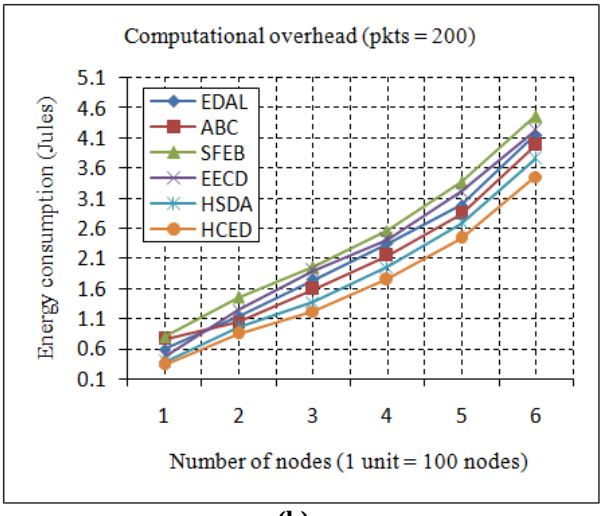


(b)

Fig. 3. Communicational overhead (a) Communicational overhead of the network when number of packets = 100 (b) Communicational overhead of the network when number of packets = 200



(a)



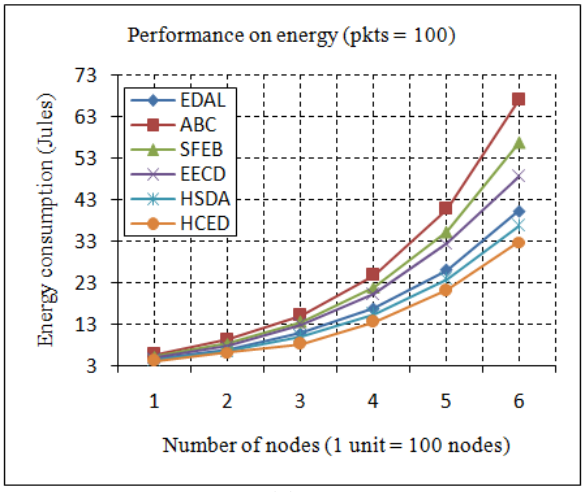
(b)

Fig. 4. Computational overhead (a) Computational overhead of the network when number of packets = 100 (b) Computational overhead of the network when number of packets = 200

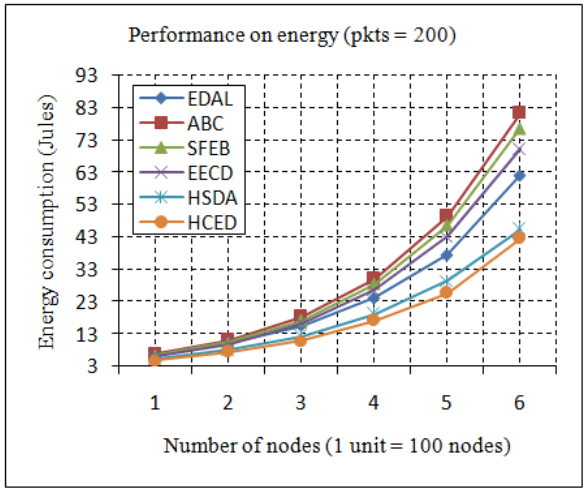
7.4 PERFORMANCE ON DELAY

Delay can be represented as the number of seconds required to aggregate the data from all the nodes in a network. The delay performance of the proposed model was evaluated and analyzed by comparing it with other models in Figure6. By varying the number of nodes from 100 to 600 and aggregating the 100 packets of data, the average delay was calculated and analyzed with other models in Figure 6(a). When aggregating 100 packets from 100 nodes, the HCED approach takes 20.32 s; if 600 nodes are involved, the HCED approach takes 44.12 s, which equates to a minimum of 6% to 24 % less delay than in other models. When aggregating 200 packets from 100

nodes, the HCED approach reduces the delay by a further 1.7% to 6.3% compared with other models. When there are 600 nodes, the HCED approach reduces the overall delay by 3.4% to 20% compared with other models. Figure 6(b) presents a detailed comparison regarding delay when aggregating 200 packets.



(a)



(b)

Fig. 5. Overall Energy consumption (a) Overall Energy consumption of the network when number of packets = 100 (b) Overall Energy consumption of the network when number of packets = 200

7.5 PERFORMANCE ON THROUGHPUT

The throughput of a network can be described as the average of successful data aggregation over a wireless communication channel. The throughput is usually measured in bits per second or data packets per time slot. The performance evaluation of the proposed model can be analyzed in terms of the number of packets successfully received by the sink node.

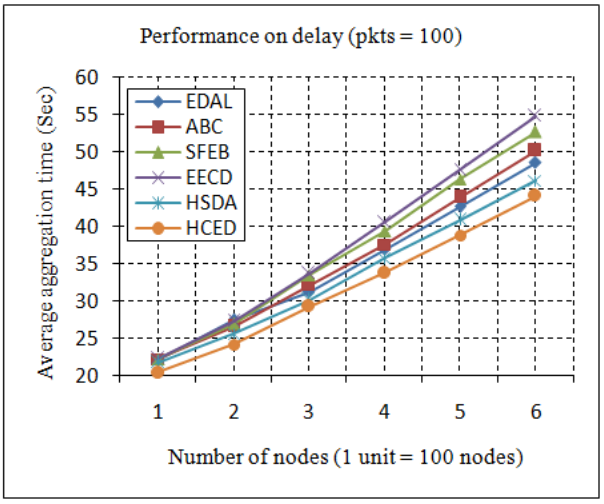
Figure 7 shows a detailed analysis of the HCED approach's throughput compared with other models.

The throughput of an HCED approach has been evaluated in two cases. In the first case, the throughput has been calculated by varying the number of nodes from 100 to 600, in increments of 100 nodes, on the static network where the node positions are fixed. When it is assumed that the sink node needs to aggregate 300 packets from the network, the simulation results in this case show that the static network with 100 nodes achieves a 99.3% throughput, which is 1% to 5% higher than existing models. When the number of nodes is increased to 600, the network achieves a 91.6% throughput, which is 3% to 9% higher than all existing models. Figure 7(a) shows the throughput performance of the HCDA regarding a static WSN. In the second case, the throughput performance of the proposed model has been evaluated in a dynamic network, in which all the nodes have mobility. When the number of nodes is 100, the proposed model achieves a 97.6% throughput, whereas it achieves 86.66% when the number of nodes is 600. In conclusion of this case, the HCDA has been shown to work well in dynamic environments, showing 3% to 8% improvement in throughput compared with existing models. Figure 7(b) describes the throughput analysis of a dynamic network with existing models.

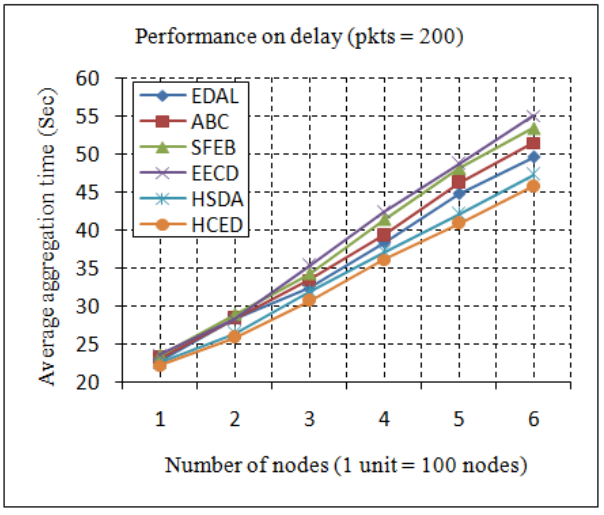
7.6 PERFORMANCE ON SECURITY

The security performance of the proposed model has been analyzed in terms of integrity and confidentiality of the aggregated data. Let consider a malicious node CN_i, which is also deployed within the network without registering at the sink node. Since node CN_i may be the neighbor of other nodes, it will also be a part of a tree construction process. When the aggregating process begins, the node CN_i will also obtain the public key, after which it encrypts the data in its own way and sends it to the sink node. Even though it uses a duplicate node ID, which is already registered with the sink, the sink node will easily identify the malicious node CN_i when decrypting the MAC for each node. This ensures the data integrity of the proposed model. Let AN_i be the attacker node that exists in the network with the intention to modify the actual data by hacking and decrypting the actual data sent among the network. Assume that node AN_i knows the public key and also have a duplicate node ID. The data aggregated by the sink node ensures its

originality by the MAC as well as the node ID of each node. The decryption algorithm used by the sink node is a secured algorithm, which cannot be hacked and implemented by any AN_i. Therefore, any node AN_i cannot decrypt any of the original data. Therefore, the proposed HCED approach ensures the data confidentiality by providing a highly secured encryption algorithm.



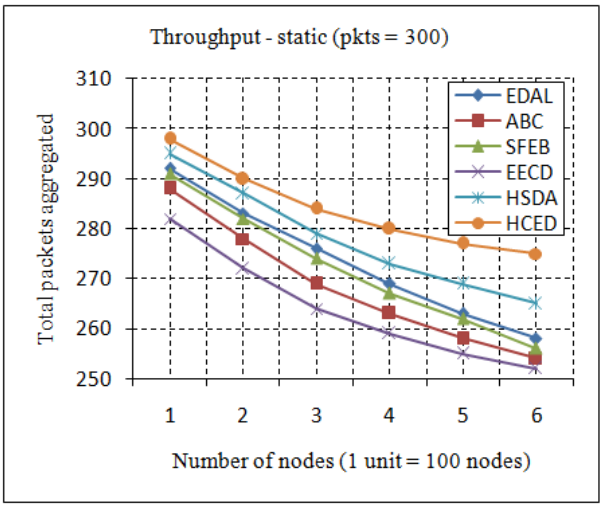
(a)



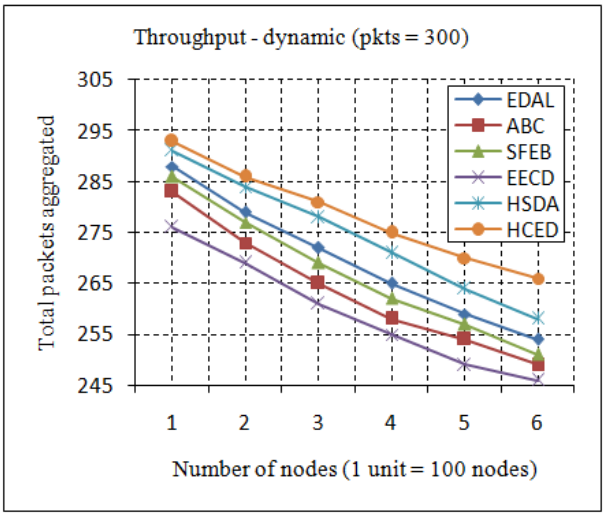
(b)

Fig. 6. Overall Delay in sec (a) Overall Delay of the network when number of packets = 100 (b) Overall Delay of the network when number of packets = 200.

The simulation results and mathematical analysis of the proposed HCED with compared to other model proves that the use of binary tree based aggregation tree provides better performance on collision avoidance than other tree structures. The implementation of hybrid aggregation algorithm which is a combination of distributed and centralized algorithm in HCED reduces the delay latency. Moreover the combination of symmetric and asymmetric cryptography technique which is used in proposed model proves the improvement in secure data aggregation than other models.



(a)



(b)

Fig. 7. Throughput (a) Throughput of the static network when number of packets = 300 (b) Throughput of the dynamic network when number of packets = 300.

8. CONCLUSION

The proposed HCED approach is an efficient data aggregation algorithm that resolves many data aggregation issues in an energy-efficient way. This research has proposed a collision-free data aggregation tree construction algorithm based on a binary search tree

protocol. To minimize the energy consumption during data aggregation, the shortest path has been identified for the tree construction. Based on the aggregation tree constructed, a delay-efficient data aggregation algorithm has been proposed to perform fast data aggregation. The secure communication has been achieved through an asymmetric key cryptography technique. The simulation results presented in this paper demonstrate that the proposed model can be implemented in static, dynamic and large-scale WSNs. Comparison with existing models further assures that the performance of the HCED approach is significantly improved in all areas regarding data aggregation issues. The authors' aim is not to highlight their research, but to point energy efficiency research in the right direction with this work. In future, researchers can improve the performance of the HCED approach so that it is suitable for extending the capabilities of WSNs with specifications regarding the "Internet of things".

CONFLICT OF INTEREST

The authors have no conflicts of interest to declare.

REFERENCES

Alshahrany, F., Abbod, M., Alshahrani, J., &Alshahrani, A. (2016). Intelligent networks data fusion web-based services for ad-hoc integrated WSNs-RFID. *International Journal of Engineering and Technology Innovation*, *6*(1), 01-15.

Boudia, O. R. M., Senouci, S. M., &Feham, M. (2015). A novel secure aggregation scheme for wireless sensor networks using stateful public key cryptography. *Ad Hoc Networks*, *32*, 98-113.

Cai, W., Chen, M., Hara, T., & Shu, L. (2010). GA-MIP: genetic algorithm based multiple mobile agents itinerary planning in wireless sensor networks. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, 1-8. IEEE.

Chakchouk, N. (2015). A survey on opportunistic routing in wireless communication networks. *IEEE Communications Surveys & Tutorials*, *17*(4), 2214-2241.

Chao, C. M., & Hsiao, T. Y. (2014). Design of structure-free and energy-balanced data aggregation in wireless sensor networks. *Journal of Network and Computer Applications*, *37*, 229-239.

Chen, M., Cai, W., Gonzalez, S., & Leung, V. C. (2010). Balanced itinerary planning for multiple mobile agents in wireless sensor networks. In *International Conference on Ad Hoc Networks* (pp. 416-428). Springer, Berlin, Heidelberg.

Chen, M., Gonzalez, S., Zhang, Y., & Leung, V. C. (2009). Multi-agent itinerary planning for wireless sensor networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 584-597. Springer, Berlin, Heidelberg.

Chen, M., Kwon, T., Yuan, Y., Choi, Y., & Leung, V. C. (2007). Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, *2007*(1), 036871.

Conan, V., Leguay, J., & Friedman, T. (2008). Fixed point opportunistic routing in delay tolerant networks. *IEEE Journal on Selected Areas in Communications*, *26*(5).

Ghosh, A., Incel, O. D., Kumar, V. A., &Krishnamachari, B. (2009, October). Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on* (pp. 363-372). IEEE.

Gopikrishnan, S., &Priakanth, P. (2016). HSDA: hybrid communication for secure data aggregation in wireless sensor network. *Wireless Networks*, *22*(3), 1061-1078.

Gupta, G. P., Misra, M., & Garg, K. (2012, January). Multiple mobile agents based data dissemination protocol for wireless sensor networks. In *International Conference on Computer Science and Information Technology* (pp. 334-345). Springer, Berlin, Heidelberg.

Han, M. K., Bhartia, A., Qiu, L., &Rozner, E. (2011, May). O3: Optimized overlay-based opportunistic routing. In *Proceedings of the twelfth ACM international symposium on mobile ad hoc networking and computing* (p. 2). ACM.

Ho, J. W., Wright, M., & Das, S. K. (2012). ZoneTrust: Fast zone-based node compromise detection and revocation in wireless sensor networks using sequential hypothesis testing. *IEEE Transactions on Dependable and Secure Computing*, *9*(4), 494-511.

Intanagonwiwat, C., Govindan, R., &Estrin, D. (2000, August). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 56-67). ACM.

Joo, C., Choi, J. G., & Shroff, N. B. (2010). Delay performance of scheduling with data aggregation in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE* (pp. 1-9). IEEE.

Karaboga, D., Okdem, S., &Ozturk, C. (2012). Cluster based wireless sensor network routing using artificial bee colony algorithm. *Wireless Networks*, *18*(7), 847-860.

Konstantopoulos, C., Mpitziopoulos, A., Gavalas, D., & Pantziou, G. (2010). Effective determination of mobile agent itineraries for data aggregation on sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, *22*(12), 1679-1693.

Li, Y., Guo, L., &Prasad, S. K. (2010). An energy-efficient distributed algorithm for minimum-latency aggregation scheduling in wireless sensor networks. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on* (pp. 827-836). IEEE.

Liu, C., & Wu, J. (2012). On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, *23*(6), 1121-1128.

Mao, X., Tang, S., Xu, X., Li, X. Y., &Ma, H. (2011). Energy-efficient opportunistic routing in wireless sensor networks. IEEE transactions on parallel and distributed systems, 22(11), 1934-1942.

Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C., &Pantziou, G. (2007, September). Deriving efficient mobile agent routes in wireless sensor networks with NOID algorithm. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on* (pp. 1-5). IEEE.

Rezvani, M., Ignjatovic, A., Bertino, E., &Jha, S. (2015). Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. IEEE transactions on Dependable and Secure Computing, 12(1), 98-110.

Rozner, E., Seshadri, J., Mehta, Y. A., &Qiu, L. (2009). SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE transactions on Mobile computing*, *8*(12), 1622.

Shakshuki, E., Malik, H., & Denko, M. K. (2008). Software agent-based directed diffusion in wireless sensor network. *Telecommunication Systems*, *38*(3-4), 161-174.

Sudarsono, A., Huda, S., Fahmi, N., Al-Rasyid, M. U. H., &Kristalina, P. (2016). Secure data exchange in environmental health monitoring system through wireless sensor network. *International Journal of Engineering and Technology Innovation*, *6*(2), 103-122.

Sun, Y., Luo, H., & Das, S. K. (2012). A trust-based framework for fault-tolerant data aggregation in wireless multimedia sensor networks. *IEEE Transactions on Dependable and Secure Computing*, *9*(6), 785-797.

Wang, X., Chen, M., Kwon, T., & Chao, H. C. (2011). Multiple mobile agents' itinerary planning in wireless sensor networks: survey and evaluation. *Iet Communications*, *5*(12), 1769-1776.

Wang, Z., Chen, Y., & Li, C. (2012). CORMAN: A novel cooperative opportunistic routing scheme in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, *30*(2), 289-296.

Xu, Y., & Qi, H. (2006). Dynamic mobile agent migration in Wireless Sensor Networks. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(1-2), 73-82.

Xu, Y., & Qi, H. (2008). Mobile agent migration modeling and design for target tracking in wireless sensor networks. *Ad Hoc Networks*, *6*(1), 1-16.

Yao, Y., Cao, Q., &Vasilakos, A. V. (2015). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, *23*(3), 810-823.

Yu, B., & Li, J. (2011). Minimum-time aggregation scheduling in multi-sink sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on* (pp. 422-430). IEEE.

Zeng, K., Yang, Z., & Lou, W. (2009). Location-aided opportunistic forwarding in multirate and multihop wireless networks. *IEEE Transactions on Vehicular Technology*, *58*(6), 3032-3040.

Zeydan, E., Kivanc, D., Comaniciu, C., & Tureli, U. (2012). Energy-efficient routing for correlated data in wireless sensor networks. *Ad Hoc Networks*, *10*(6), 962-975.