Computer Networks 160 (2019) 51–64

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Scheduling of data aggregation trees using Local Heuristics to enhance network lifetime in sensor networks

Preeti A. Kale*, Manisha J. Nene

Department of Computer Science and Engineering, Defence Institute of Advanced Technology, Defence Research and Development Organization, Pune 411021, India

ARTICLE INFO

Article history:
Received 6 August 2018
Revised 27 April 2019
Accepted 20 May 2019
Available online 25 May 2019

Keywords:
Local Heuristics
Network lifetime
Data aggregation trees

ABSTRACT

Data gathering is a basic requirement in many applications of Wireless Sensor Networks (WSNs). In tree based data gathering, Data Aggregation Tree (DAT) is constructed by the sink or by the nodes in a distributed manner. In this paper, we study the problem of enhancing Network Lifetime (NL) using hybrid DAT construction methods. In hybrid methods of DAT construction, the sink and the nodes collaboratively construct the DAT. We propose three algorithms for Scheduling DATs using Local Heuristics with Ordering (SDLHO), with Randomization (SDLHR) and with Tree factor (SDLHT) techniques.These techniques avoid disparity in energy levels of the nodes and increase the survivability of the network. In addition, to address imperfect link quality, we propose an algorithm for Scheduling DATs using Local Heuristics with Ordering based on Link Quality (SDLHO-LQ). Rigorous simulation results demonstrate the efficacy of the proposed algorithms; and their ability to scaleup to suit deployment of applications in harsh regions. Further, their performances evaluated to quantify the amount of enhancements of NL with the existing state of art is propitious to suit the distributed environments.

1. Introduction

In WSNs, the sensors gather and send data to the sink. In applications like environmental monitoring, battlefield surveillance, structural health monitoring, pipeline monitoring and precision agriculture, the sensor nodes are typically randomly deployed and left unattended. Transmission of packets between sensors consumes energy. In terms of power consumption, transmitting a single bit of data is equivalent to 800 instructions [1]. In such situations, employing data gathering mechanisms that judiciously utilize battery power of sensor nodes is essential. By combining data packets from different sensor nodes, the number of packet transmissions is reduced. This technique of combining data so that crucial data is made available at the sink is termed as in-network data aggregation DA.

In-network DA using tree based routing structure saves the cost of maintaining a routing table at each node and is suitable in energy constrained WSNs. Tree based DA reduces the number of packet transmissions, decreases energy consumption and improves NL. However, reducing packet transmissions is a challenging problem as it depends on the amount of data generated at each node and the structure of the DAT.

In tree based DA, the DAT can be constructed in two ways. (1) In centralized DAT construction, the sink gathers information of the entire network and then constructs a DAT using a suitable tree construction algorithm [2–7]. (2) In distributed method of DAT construction, the nodes communicate with their neighboring nodes and select appropriate parent and child nodes to construct a DAT [1,5,8–10]. In this case, the sink does not require information about the entire network however this method adds communication overhead. In a DAT, each leaf node senses data and transmits it to its parent node while an intermediate node senses data, receives data from its child nodes, aggregates data and then transmits data to its parent node. Data generated by all nodes reaches the sink per unit time and is considered as one round of data collection. NL is measured in terms of maximum number of rounds of data collection in the network until the network partitions.

For example, Fig. 1(a) represents a sensor network with eight sensor nodes and a sink. Let Fig. 1(b) and Fig. 1(c) represent two different DAT structures $T1$ and $T2$ for the network. Let each node transmit a data packet in each data collection round and let data packet be of fixed size. Let the energy consumption for transmitting and receiving one data packet at a node be 1 unit. If the initial energy ϵ of each node is assumed to be 10 units, then the maximum number of rounds of data collection for $T1$ are 3 and denotes

* Corresponding author.
E-mail addresses: preeti_pcse16@diat.ac.in, kalepreeti@gmail.com (P.A. Kale), mjnene@diat.ac.in (M.J. Nene).

https://doi.org/10.1016/j.comnet.2019.05.017

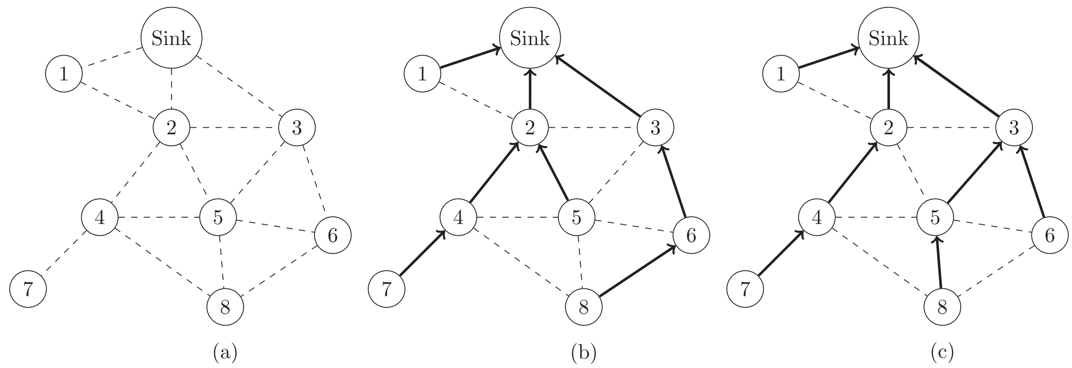

Fig. 1. Example: (a) A wireless sensor network with eight nodes and a sink (b) DAT $T1$ (c) DAT $T2$.

NL. Similarly for $T2$, NL is 3. However let us assume that $T1$ works for 2 rounds after which the DAT is reconstructed to $T2$. $T2$ then works for another 2 rounds resulting in a total of 4 rounds of data collection. Scheduling $T1$ for 2 rounds followed by $T2$ for 2 rounds is better than using only $T1$ or only $T2$ in terms of NL.

This motivates us to investigate the problem of scheduling a sequence of DATs in distributed environments. The sequence of DATs work in different data collection rounds so that the NL is maximized. We term this problem as Tree Reconstruction based Scheduling to improve network Lifetime (TRSL).

Following are the major contributions

- TRSL problem is addressed using heuristics that employ NL-improving strategies.
- NL is enhanced using hybrid DAM model in which both the sink and the individual sensor nodes collaboratively construct the DAT.(Ref Section 2.2)
- Algorithms for the TRSL problem in idealistic scenarios are proposed and compared. The emphasis is on the suitability of the algorithm to cater to application needs.
- The uncertainty in realistic scenarios posed by environmental and hardware based factors is tackled by using appropriate heuristics.

The algorithms are evaluated and compared with rigorous simulations. Results show that they balance energy consumption of the nodes and significantly improve NL.

1.1. Related work

Maximizing network lifetime and minimizing energy consumption cost are extensively researched problems in sensor networks. The two problems are related since network lifetime can be enhanced by reducing energy consumption cost. In energy minimization problems, the objective is to consume minimum energy in each data collection round while in network lifetime maximization problems, the objective is to extend the number of data collection rounds for which the network can survive.

In these problems the network is modelled as a graph and energy efficiency is achieved by constructing DATs. DAT with minimum energy cost is a minimum spanning tree if all incoming data at each node is aggregated into a single packet [3]. However, when networks have nodes with different capabilities and incoming data is aggregated into multiple packets, the minimum energy cost DAT is modelled as a Minimum Steiner Tree problem. This problem is proven to be NPComplete in [11]. In [12], Minimum Fusion Steiner Tree (MFST) is proposed for energy efficient data gathering and in [2], MECAT_RN, an approximate tree construction algorithm is devised. In [13], correlation aware DAT is constructed. Heuristics based approximation algorithm called as Balanced SPT/TSP and Leaves Deletion algorithms are proposed in [14]. Adaptive Fusion Steiner Tree in [15] constructs DAT by adapting routing tree using single packet and multiple packet based aggregation. In [16], minimum energy cost data aggregation scheduling aggregates fixed number of data into a packet.

DATs constructed by energy minimization algorithms tend to deplete the energy of certain nodes that lie on the optimal communication path and eventually results in partitioned network. Prolonging the number of data collection rounds of the network the objective of the proposed work and aims at maximizing NL.

DAT with maximum NL is a Minimum Degree Spanning Tree known to be NP Complete [17]. Several techniques in [3–5,8–10,18–20] maximize NL using DATs in which intermediate nodes aggregate incoming data into a single packet. Assuming that each node can aggregate all incoming data into a single packet is not suitable in practical conditions. Hence in [17,21,22] DATs are constructed with intermediate nodes aggregating incoming data into multiple packets. In these cases a fixed number of data are allowed to be aggregated into one packet as fixed size packets avoid additional overhead and resource management costs involved in variable packet sizes [23,24]. In [17], each node produces single unit of data whereas in [21,22] data size is of variable units. Energy Conserving Routing Tree and Local Optimization are heuristic algorithms discussed in [17] that improve NL. In [21] routing metric of rate of energy increase is considered. In [22], Local Tree Reconstruction Based Scheduling Algorithm (LTRBSA) improves NL using heuristics. Tree reconstruction with path reestablishment is discussed in [25]. These algorithms require knowledge about network topology and have robustness and scalability limitations.

An approximation algorithm to balance the number of child nodes and prolong NL is proposed in [26]. In [27], NL is maximized using genetic algorithms. A fuzzy logic based routing for NL enhancement is demonstrated in [28] and technique for maximum NL DAT using transmission power levels is developed in [29]. An exhaustive survey in [30] studies NL maximization techniques.

Most of the studies mentioned above consider idealistic communication links in the network. In idealistic scenario, nodes that are within each other's communication range, always communicate successfully. However in reality, environmental and hardware related factors induce uncertainty in communication, hamper the link quality and affect network performance.

The uncertainty in communication links is due to imperfect radio connectivity caused by interference, multipath propagation and distortion in radio signals [8,31–39]. In real deployments, the transmission range has three regions as connected, disconnected and transitional region [31,40]. The links in the transitional region induce link uncertainty, since the extent and significance of these links shows large variation as observed in [36].

A detailed survey to understand the fundamentals associated with Link Quality Estimation(LQE) and its classification is presented in [36,37]. Link Quality Estimators (LQEs) are broadly classified into two categories as hardware-based and software-based LQEs [36,37]. The software-based LQEs are classified into three categories as Packet Reception Ratio (PRR) based, Required Number of Packet Transmissions (RNP) based and score based. The software-based LQEs provide a fine grain link quality estimation but need additional computation [36]. In hardware-based LQEs, the link quality is estimated by using Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) value which is provided by the radio chips. The hardware-based LQEs are directly read from the radio transceiver and they do not require any additional computation [36].

In [38], methods to estimate link quality using RSSI, LQI and PRR are analysed for large scale indoor settings. The impact of different environmental conditions and hardware settings on the RSSI, LQI and PRR values is described in [41]. Data driven prediction models to estimate link quality using RSSI, LQI and PRR are analysed in [42]. In [8,43], a probabilistic network model is incorporated to address lossy link behavior and a DAT based NL maximization algorithm is devised. In [44], an energy efficient link quality based DAT is presented.

In [35,36,38], it is reported that LQI classifies link quality more accurately as compared to RSSI. In addition, LQI has a greater correlation with link delivery ratio and packet error ratio as compared to RSSI [36,37]. Hence, to handle uncertainty in communication links in realistic scenarios, link quality is modelled using LQI in the proposed work.

The proposed work enhances NL by scheduling a sequence of DATs using centralized as well as distributed DAT construction techniques that are scalable and where each intermediate node aggregates incoming data into multiple packets. In addition, the proposed work uses LQI to address link uncertainty.

1.2. Outline of the paper

The remainder of this paper is organized as follows. In Section 2 the required notations and models are presented followed by TRSL problem statement and contributions. The proposed algorithms Scheduling DATs using Local Heuristics with Ordering SDLHO algorithm and Scheduling DATs using Local Heuristics with Randomization SDLHR are discussed and presented in Section 3. To address uncertainty in WSNs, Scheduling DATs using Local Heuristics with Link Quality algorithm is devised in Section 3.7. In Section 4, the proposed Scheduling DATs using Local Heuristics with Tree factor SDLHT algorithm is presented. Using simulations the performance of the proposed algorithms is evaluated in Section 5. Finally, Section 6 concludes the paper.

2. Proposed work

In this section, the network model and data aggregation models are described and representation for scheduling a sequence of DATs is discussed. The problem statement is enumerated and contributions are mentioned.

2.1. Network model

The WSN has N sensor nodes randomly deployed in the area of interest. Node s in the network represents a sink node. The sink node has sufficient resources required for communication and computation. It initiates communication in the network and receives data from N sensor nodes.

The network is modelled as a Graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, such that $\mathcal{V}$ represents N sensor nodes and a sink node s given as $|\mathcal{V}| = N + 1$. It is assumed that all nodes have equal transmission range g. Each node u can communicate with the sink s either through single or multihop communication and each node is in the communication range of at least one node. $NB(u)$ denotes the set of neighboring nodes of a node u.

$\mathcal{A}$ is the set of communication links in the network. The communication links are modelled such that each edge $(u, v) \in \mathcal{A}$ has an associated link weight $w_{(u,\,v)}$ that determines successful communication between nodes u and v. The two approaches used for modelling links, represent idealistic and realistic communication links as follows.

For each edge $(u, v) \in \mathcal{A}$

$$w_{(u,v)} = \begin{cases} 1 & \text{perfect links in idealistic scenario} \\ wl & \text{imperfect links in realistic scenario} \end{cases} \tag{1}$$

In idealistic scenario, link weight value of one denotes perfect link quality. In this case, no node failure due to environmental factors are considered and packet collisions and transmission congestion do not affect the proposed work.

In realistic scenario, the uncertainty in communication links is modelled as imperfect links. In this case, $w_{(u,\,v)}$ denotes the estimated link quality for communication between nodes u and v. In the proposed work, weight wl indicates the LQI value of the link. LQI values are obtained from experiments conducted in open concrete field in [35].

Sensor nodes in the network are either relay nodes or source nodes. Let S be a set of source nodes and R be a set of relay nodes then $N = |S| + |R|$ and $\mathcal{V} = \{S \cup R \cup s\}$. For any sensor node $u \in \{S \cup R\}$, δ_u is data bits generated, I_u is incoming data, Z_u is total incoming and generated data given as $Z_u = I_u + \delta_u$ and O_u is outgoing data. A node u is a source node if it generates $GN(\delta_u)$, receives $RC(I_u)$, aggregates $AG(Z_u)$ and transmits $TR(O_u)$ data and this functionality is given as $f(u|u \in S) = (GN(\delta_u), RC(I_u), AG(Z_u), TR(O_u))$, where $\delta_u > 0$. A relay node u performs all functions of a source node except for generation of data and has $\delta_u = 0$. Functionality of relay node u is represented as $f(u|u \in R) = (RC(I_u), AG(I_u), TR(O_u))$.

2.1.1. Data Aggregation Tree

DAT is modelled as a spanning tree $T = (V_T, E_T)$ of $\mathcal{G}$ such that $V_T = \mathcal{V}, E_T \subseteq \mathcal{A}$ for a network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. Root node of T represents the sink node s. In the proposed work, T is constructed such that each node u can communicate with the sink in minimum number of hops given by hop count h_u. Sink has hopcount 0. Hop count represents the depth of a node in T. Nodes $v \in V_T \backslash s$ at depth dp have parent nodes at depth $dp - 1$. Let nodes u and v be two nodes such that $h_v = h_u - 1 \wedge v \in NB(u)$ then v is a valid parent node of node u. The set of valid parent nodes of a node u is represented as $VP(u)$.

The structure of DAT determines I_u, Z_u, O_u values at each node u. For node u in T, let $C(T, u)$ denote the set of child nodes of u and c_u be parent node of u. Each node receives data from child nodes and forwards to its parent node. Then the values of I_u and Z_u are computed as follows

$$Z_u = I_u + \delta_u \tag{2}$$

$$I_u = \sum_{v \in C(T,u)} Z_v \tag{3}$$

Amount of outgoing data O_u at a node u depends on the application. If application requirement is to generate and send data continuously to the sink then aggregation of data at intermediate nodes is not performed. In this case, each node u transmits its total received and generated data to its parent node c_u and $O_u = Z_u$. If the application requires that data from all nodes be sent to sink per unit time t, then each node senses and accumulates data and the accumulated data is combined and sent per time t.

Combining data is based on the aggregation ratio α which is the ratio of the amount of outgoing data to that of incoming data given as $\alpha = \frac{O_u}{Z_u}$. Lower values of α indicate that more amount of data is combined at the node and results in reduced number of transmissions. Aggregation ratio α is also defined as the number of data units that can be combined into one packet [2,21,22,45] and is given as

$$P_u = \left\lceil \frac{Z_u}{\alpha} \right\rceil \tag{4}$$

where P_u is number of unit size packets forwarded by node u. In this case, higher values of α indicate that less number of packet transmissions are required. In the proposed work aggregation ratio computation is based on Eq. (4).

2.1.2. Data aggregation round

Each node u generates data δ_u per epoch and reports it to the sink where epoch represents a time interval. To facilitate data reporting, each node in DAT T decides the duration of interval during which it will report data. In the proposed work, this duration of interval is based on maximum hopcount in T. Let hx represent the maximum hopcount in T, given as $hx = max\{h_1, h_2, \ldots, h_N\}$. Nodes at depth hx are leaf nodes and forward data to nodes at depth $hx - 1$. These nodes in turn forward data to nodes at depth $hx - 2$ and this process continues until data from all nodes reaches the sink.

Let interval [0, t_n] represent one epoch of t_n seconds. Using hx, the duration of interval i for data reporting is set to $i = \lfloor \frac{t_n}{hx} \rfloor$. Then the duration of interval for reporting data for any node at depth dp other than sink is given by $[t_n - (dp \times i), t_n - (dp \times i) + i]$. In this way all nodes report data to the sink in an epoch of [0, t_n] seconds. Data from all nodes being reported to the sink per epoch is a data aggregation round.

2.1.3. Energy consumption cost estimation

Initial energy of all nodes $u \in \mathcal{V} \setminus \{s\}$ in the network is the same and is represented as ϵ units. The residual energy at a node u is the energy available at that node given by ϵ_u. Initially $\epsilon_u = \epsilon$ for all nodes. In each data aggregation round, each node u consumes energy for communication, computation and sensing given by $\mathcal{E}_{cn_u}, \mathcal{E}_{cp_u}, \mathcal{E}_{sn_u}$. Total energy consumed per data aggregation round for a node u is $\mathcal{E}_u$ such that $\mathcal{E}_u = \mathcal{E}_{cn_u} + \mathcal{E}_{cp_u} + \mathcal{E}_{sn_u}$.

The energy cost of computation is proportional to the amount of data α aggregated at a node, as this data needs to be scanned at least once before any computation [46]. Larger values of α imply more number of instructions required to be executed at the node and hence involves higher energy cost of computation. Let $f(\alpha)$ be an increasing function bounded by α that determines $\mathcal{E}_{cp_u}$ such that

$$\mathcal{E}_{cp_u} = f(\alpha) \tag{5}$$

When $\alpha = 1, \mathcal{E}_{cp_u} = \Omega(1)$ is the best case for $\mathcal{E}_{cp_u}$. For $\alpha = Z_u, \mathcal{E}_{cp_u} = O(\alpha)$ is the worst case for $\mathcal{E}_{cp_u}$. Let the energy cost of communication $\mathcal{E}_{cn_u}$ be y times the energy cost of computation $\mathcal{E}_{cp_u}$. Then,

$$\mathcal{E}_{cn_u} = y \times \mathcal{E}_{cp_u} \tag{6}$$

The value of y depends on the underlying platform. Based on literature survey study, it is observed that for TinyOS motes, $y \approx 800$ [1], for Tmote Sky, $y \approx 1000$ [47], for WINS NG 2.0 nodes $y \approx 1400$ [48] and for mica2 motes $y \approx 1000$ [49]. Let $y \approx 800$.

In the proposed work, it is assumed that $\alpha \ll 800$. Using Eqs. (5) and (6), it is inferred that $\mathcal{E}_{cp_u} \ll \mathcal{E}_{cn_u}$. Hence the energy cost of computation is considered as insignificant and the energy consumption cost is estimated based on energy required for communication given as $\mathcal{E}_u = \mathcal{E}_{cn_u}$[2,4,6,22,50].

In a DAT, the energy $\mathcal{E}_u$ consumed by each node u is based on number of packets received from its child nodes and number of packets transmitted to its parent node. Let Tx represent energy required for transmitting a data packet to its parent node and Rx represent energy required for receiving a data packet from a child node. Because the transmission energy cost is about double the reception energy cost, it is assumed that $Tx = 2$ and $Rx = 1$ [2,4,6,22,50]. As a result, $\mathcal{E}_u$ is given as.

$$\mathcal{E}_u = Rx \sum_{v \in C(T,u)} P_v + Tx.P_u \tag{7}$$

As $\mathcal{E}_u$ is the energy required by node u for one DA round, the maximum number of DA rounds for which u can survive is calculated as

$$L_u = \left\lfloor \frac{\epsilon}{\mathcal{E}_u} \right\rfloor \tag{8}$$

L_u is the lifetime of node u in terms of DA rounds. The residual energy at node u after L_u rounds is given by

$$\epsilon_u = \epsilon - (L_u \times \mathcal{E}_u) \tag{9}$$

After L_u rounds, $\epsilon_u < \mathcal{E}_u$ and u does not have enough residual energy to report its data. Such nodes that cannot report data are termed as low residual energy node in the DAT. The estimation of energy consumption cost of all nodes in the network determines NL. Let L be set of lifetimes of N nodes given by $L = \{L_1, L_2 \ldots L_N\}$ then the lifetime of the network is the lifetime of the minimum lifetime node in the network represented as D such that

$$D = min\{L_1, L_2, \ldots, L_N\} \tag{10}$$

In the proposed work, if a DAT is denoted as T_k, then the lifetime of T_k is represented by D_k. The minimum lifetime node is termed as the bottleneck node bn in the proposed work. Using Eq. (10),

$$L_{bn} = D \tag{11}$$

2.2. Data Aggregation Models

Data Aggregation Models (DAM) investigate the role of individual nodes and sink in DAT construction.

2.2.1. Centralized DAM

In this model the sink node has knowledge about all the sensors in the network and computes DAT using algorithms like shortest path tree. After formation of DAT T at sink node, the sink sends parent and child node data of each node u, to the respective node u in the network. Accordingly, u starts receiving data from $C(T, u)$ and transmitting to c_u. Sink node knows all paths in the network and we call sink as the decision maker in the network.

2.2.2. Distributed DAM

In this model the sink broadcasts a message asking nodes to organize into a routing tree. The message includes the sending node sn and hop count hc and is represented as (sn, hc). As hop count of the sink is zero, it sends message $(sn = s, hc = 0)$. Each node v receiving this message sets its own hop count $h_v = hc + 1$ and sets its parent node c_v as the sending node in the message. Node v then sends acknowledgement message $(sn = v)$ to its parent node c_v so that c_v updates its set of child nodes C with v. The node then rebroadcasts the message with its data $(sn = v, hc = h_v)$. The process continues until all nodes receive the message and a DAT T is constructed. Each node u in T decides its $C(T, u)$ and c_u. Each node u is a decision maker as it computes $\mathcal{E}_u$ to decide the path along which data is sent. Sink node only knows its neighboring nodes in the network.

2.2.3. Hybrid DAM

In Hybrid DAMs both the sink and the individual sensor nodes collaboratively construct the DAT. In Hybrid DAM I, DAT T is initially constructed in a distributed manner as in Distributed DAM. It is assumed that sink has knowledge of its k-hop neighborhood which are nodes within k-hop distance from the sink represented by $G = (V, E)$. DAT T_k termed as k-hop subtree is computed by the sink such that $T_k = (V_T, E_T)$, $V_T = V, E_T \subseteq E$. For each node $u \in V_T$, the sink sends updated parent and child node data $C(T_k, u)$ and c_u to the respective node u in the network to facilitate communication. In this DAM, each node u decides its communication path based on its value of $\mathcal{E}_u$. As sink knows its k-hop neighborhood, it evaluates if reconstructing T_k improves L_{bn} for the network. In this DAM, both the sink and the nodes decide the path for sending data and is termed as hybrid DAM I.

In Hybrid DAM II, after the initial distributed DAT construction, set of nodes within k-hop distance from the sink V_k reorganize the tree in a distributed manner. Each node $u \in V_k$ analyses its neighboring nodes and refines and reconstructs the initial paths depending on the application. In this case all nodes are path establishment decision makers for initial DAT while nodes within k-hop distance from the sink refine decisions for path establishment without the sink knowing about it.

The proposed work considers Hybrid DAM I and II

2.3. Scheduling data aggregation trees

Different DATs are established in turns at different time periods in the network. Time period is measured in terms of data aggregation rounds. A schedule refers to planning which DAT T will work for how many rounds d and is based on heuristics applied at each node in T. In the proposed work, heuristics depends on the neighboring node information only and we call it Local Heuristics. Schedule U is a composite variable with two components $List_1$ and $List_2$ and represented as $U = \{List_1\}, \{List_2\}; \{List_1\} = \{T_1, T_2, \ldots, T_t\}, \{List_2\} = \{d_1, d_2, \ldots, d_t\}$. Here $\{T_1, T_2, \ldots, T_t\}$ is a sequence of DATs and $\{d_1, d_2, \ldots, d_t\}$ a list of their respective number of DA rounds. The schedule represents that DAT T_1 works for d_1 DA rounds followed by DAT T_2 working for d_2 DA rounds, respectively.

For example, consider the graph in Fig. 1(a) which represents a network of 8 sensor nodes and a sink node. It is assumed that source nodes sense temperature values and forward to the sink. MAX temperature value is required at the sink every minute and can be gathered in following two ways

2.3.1. Scenario 1

In this scenario, the structure of DAT $T1$ in Fig. 1(b) determines how data is gathered from the network. Each intermediate node u in $T1$ computes MAX of its received and generated data and sends to c_u. The process continues until $node\_1, node\_2, node\_3$ transmit MAX values from their respective subtrees to the sink. Sink then computes the MAX of the 3 received values. The process of computing MAX temperature in the network continues as long as all nodes have enough energy to report data. The maximum number of data reporting DA rounds determines NL.

2.3.2. Scenario 2

In this scenario, DAT $T1$ may be reconstructed to $T2$ as shown in Fig. 1(c) and these trees together determine how data is gathered from the network. For example, $T1$ computes MAX temperature in the network for first few rounds. Based on energy consumed by the nodes, $T1$ is reconstructed to say $T2$. Using $T2$, MAX temperature is determined in the next few rounds.

The proposed work determines how data can be gathered from the network using a sequence of DATs so that the network lifetime is improved.

2.4. TRSL problem statement

Given

1) a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ representing the network
2) $\mathcal{V} = \{S \cup R \cup s\}$. S is set of sources, R is set of relay nodes and s is sink. $|\mathcal{V}| = |S| + |R| + 1 = N + 1$.
3) $\delta_v \in Z^+$ for each source $v \in S$ and $\delta_v = 0$ for $v \in R$.
4) $\epsilon_v \in Z^*$ is residual energy,
5) an aggregation ratio $\alpha \in Z^+$,
6) energy for transmitting a packet $Tx \in R^+$ and for receiving a packet $Rx \in R^+$,

The objective of Tree Reconstruction based Scheduling to improve network Lifetime TRSL problem is to find a schedule $U = \{T_1, T_2, \ldots T_t\}, \{d_1, d_2, \ldots, d_t\}$ that maximizes the network lifetime given as

$$Max \sum_{i=1}^{t} d_i \tag{12}$$

such that $\{D_1, D_2, \ldots, D_t\}$ represent network lifetime for DATs $\{T_1, T_2, \ldots T_t\}$ and $d_1 \le D_1, d_2 \le D_2 \ldots d_k \le D_t$

Further, this problem statement is extended to suit two cases I and II for scenario 1 and 2 (Refer Sections 2.3.1 and 2.3.2), respectively

Case I: $d_1 = D_1, d_2 = D_2 \ldots d_t = D_t$
Case II: $d_1 \le D_1, d_2 \le D_2 \ldots d_t \le D_t$

2.5. Contributions

The following are the contributions.

1) Proposed Local Heuristics based on the residual energy of neighboring nodes for DAT construction. This heuristics for TRSL is implemented and demonstrated using the proposed algorithm Scheduling DATs using Local Heuristics with Ordering (SDLHO). SDLHO is implemented using hybrid DAM I (Ref. Section 2.2.3). Performance of SDLHO is evaluated with rigorous experiments and results show that SDLHO is efficient when scaled up and improves NL by 50%. (Ref. Section 5.4).
2) Proposed lightweight, randomization based Local Heuristics for DAT construction. This heuristics for TRSL is implemented and demonstrated in Scheduling DATs using Local Heuristics with Randomization (SDLHR) algorithm. This technique reduces computation overhead at a node and enhances NL by 40%, as shown by rigorous experimentation results. (Ref. Section 5.4).
3) To address the concerns regarding uncertainty in communication links in practical and realistic situations, Scheduling DATs using Local Heuristics with Ordering based on Link Quality (SDLHO-LQ) algorithm is proposed. SDLHO-LQ assures best possible link quality using LQI values. Rigorous simulation results show Network Liftime (NL) values observed due to uncertainty in communication links (Ref. Section 3.7).
4) A new basis for tree reconstruction termed tree factor is investigated and proposed to overcome the challenges associated with hybrid DAM I. Tree factor based TRSL technique is implemented and demonstrated by Scheduling DATs using Local Heuristics with Tree factor (SDLHT) algorithm. SDLHT employs hybrid DAM II (Ref. Section 2.2.3) and is computationally efficient. (Ref. Section 5.7).
5) The proposed algorithms SDLHO, SDLHO-LQ, SDLHR and SDLHT are scalable and improve NL as compared to existing state of art by 5 – 50% for deployment of 100 – 1000 nodes, respectively in common platform settings. Performance is

evaluated with 30 simulation runs and results show consistent performance. (Ref. Table 2).

3. Algorithm SDLHO: Scheduling DATs using Local Heuristics with Ordering

SDLHO algorithm is discussed and proposed in Sections 3.1 to 3.4. SDLHO with example is discussed in Sections 3.5 and 3.6 presents its computational complexity. In Section 3.8, SDLHR algorithm is proposed.

SDLHO algorithm (Algorithm 1) employs hybrid DAM I. In SDLHO (Fig. 2), the initial DAT is constructed in a distributed manner while all other steps are executed at the sink node.

3.1. SDLHO: Initial distributed DAT construction

Sink node initiates communication in the sensor network by broadcasting messages to form a DAT. Broadcast message is represented as (sn, hc) where sn represents the node sending the message and hc is the number of hops required by the sending node to reach the sink. Initial message broadcasted by s is $(sn = s, hc = 0)$. Neighboring nodes $NB(s)$ receive the message. Each $v \in NB(s)$ sets its own hopcount value $h_v = hc + 1$, sets its parent node $c_v = sn$ and adds sn to its $NB(v)$. Node v then updates message to $(sn = v, hc = h_v)$ and broadcasts to its neighboring nodes $NB(v)$. If a node v receives multiple messages, then the message with minimum value of hc is considered by the node for setting its h_v and c_v values. This process is repeated until all nodes v in the network receive a message and set their h_v, c_v and $NB(v)$ values.

Each node v checks its $NB(v)$ to find maximum hopcount value amongst its neighbors given as $hm = max\{h_{u_1}, h_{u_2} \dots h_{u_k}\}, u_i \in NB(v), i = 1.k$. If $h_v \geq hm$ then v is identified as a leaf node. Each leaf node v sends an acknowledgement message represented as $(sn = v)$ to its parent node c_v. On receiving the message, c_v adds v to its child node set C. This process is repeated by all nodes until the message reaches the sink and the DAT T_{isp} is constructed. T_{isp} is a shortest path tree as each node can reach the sink in minimum number of hops.

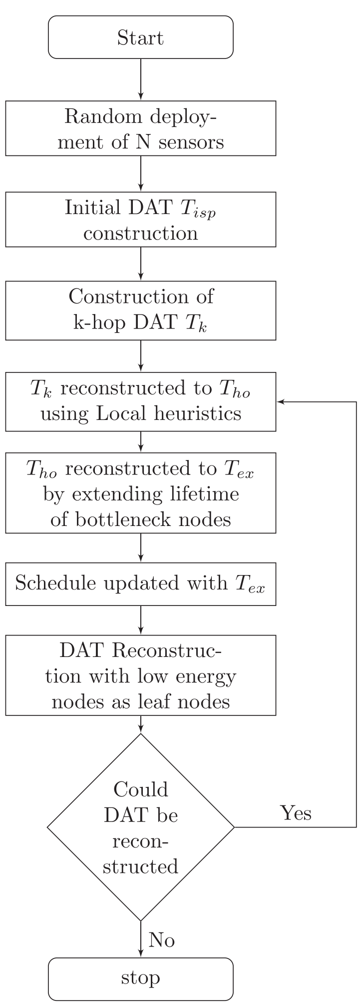

Fig. 2. Flowchart for SDLHO.

3.2. SDLHO: Local Heuristics

In the proposed work, the initial DAT is reconstructed by using heuristics. Each node applies heuristics using information about neighboring nodes only and is termed as Local Heuristics. Each node assesses its parent node to explore suitability of changing the parent node. Changing parent node termed as parent switching is performed only if it improves the lifetime of its parent node.

The steps are described in the function *Local_Heuristics* as follows. Each node u checks its valid parent list $VP(u)$. The nodes in $VP(u)$ are ordered in decreasing order of their lifetimes. Higher lifetime nodes demonstrating more residual energy are checked first for a possible parent switch. Node u switches its current parent node c_u if parent switching improves lifetime L_{c_u}.

3.3. SDLHO: Extending lifetime of bottleneck node

In the proposed work, the sink node identifies the bottleneck node in k-hop subtree T using Eq. (11). Extending lifetime of the bn improves NL.

The lifetime L_{bn} of the bottleneck node bn is improved by applying two strategies. First strategy is based on examining if the number of child nodes of bn can be reduced. Reducing number of child nodes saves energy required to receive packets and extends lifetime of bn. If this strategy fails to improve L_{bn} then a second strategy is applied. In this case, the minimum lifetime node m amongst child nodes of bn is identified as the new bottleneck node and the first strategy is reapplied. The process continues until node with distance k from the sink is reached.

3.4. SDLHO Algorithm

Algorithm 1.

3.5. SDLHO Example

A randomly deployed sensor network $G = (V, E)$ of 60 nodes with node 61 as sink node is shown in Fig. 3. It is assumed that $\alpha = 3$, $Rx = 1$, $Tx = 2$ and $k = 3$.

Steps

1) DAT T_{isp} (Fig. 4) is constructed and each node computes $Z_u, P_u, \mathcal{E}_u, L_u$ from Eqs. (2), (4), (7) and Eq. (8), respectively.
2) A k-hop subtree T (Fig. 5) is constructed. The nodes in T and their δ_u values are shown in first and second columns of Table 1. Column $\mathcal{E}_u[T]$ in the table represents the computed $\mathcal{E}_u$ values for T. Using Eq. (10), value of D is computed to be 12 rounds.
3) Fig. 6 shows the reconstructed tree T_{ho}. *nodes*_6, 25, 30, 34, 41, 43, 44 and *node*_49 switch to new parent nodes *nodes*_52, 45, 18, 19, 34, 45, 18 and *node*_18, respectively as shown. Fourth column in Table 1 represents $\mathcal{E}_u$ for new DAT T_{ho}. From Eq. (10), $D_{ho} = 28$ rounds.

Algorithm 1 SDLHO.

Input: A randomly deployed WSN as graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with N sensor nodes and a sink node s. For each node $u \in \mathcal{V}$, δ_u and ϵ_u represent data generated by the node and its initial energy. Tx, Rx is energy required for transmitting and receiving a packet respectively. k represents distance in terms of number of hops from sink s.
Output: A schedule $U = \{T_1, T_2, ...T_t\}, \{D_1, D_2, ..D_t\}$
Steps

1. Construct Distributed DAT $T_{isp}(V_{isp}, E_{isp})$ such that $V_{isp} = \mathcal{V}, E_{isp} \subseteq \mathcal{A}$. Each node $u \in V_{isp}$ computes $Z_u, P_u, \mathcal{E}_u, L_u$ using Eq. (2),(4),(8) (10) respectively.
2. Identify nodes within k-hop distance from s. Represent these nodes and their connecting edges as graph $G = (V, E)$. Construct a spanning tree $T = (V_T, E_T)$ termed as k-hop subtree of T_{isp} such that $V_T = V, E_T \subseteq E$. Compute D of T using Eq. (10).
3. Reconstruct T to T_{ho} by executing *Local_Heuristics*
4. Find bottleneck node bn in T_{ho}. Reconstruct T_{ho} to T_{ex} by executing *Extend_L_bn*. *Extend_L_bn* extends the lifetime of subtree rooted at bn. Repeat this step for T_{ex} until T_{ex} cannot be reconstructed.
5. Update schedule U by adding T_{ex}, D_{ex}. Update ϵ_u for all nodes using Eq.(9).
6. Execute *Tree_with_Y_leafnodes* to construct T_l such that low residual energy nodes are leaf nodes. If T_l is constructed successfully then repeat steps 3 to 6 for T_l else return schedule U.

function *Local_Heuristics*
Input: $T, D, G(V, E)$
Output: T_{ho}, D_{ho}

1. for each node $u \in V$do
2. $[P, T_p]$=*Parent_Switching*(u, T)
3. if $P == 1$ // T updated to T_p
4. $T_{ho} = T_p$
5. Compute D_{ho}
6. return (T_{ho}, D_{ho})
7. else continue
8. end for
9. return (T_{ho}, D_{ho})

function *Extend_L_bn*
Input: $bn, T_{ho}, D_{ho}, G(V, E)$
Output: P, T_{ex}, D_{ex}

1. $P = 0$
2. for each node $u_i \in C(T_{ho}, bn)$ do
3. $[P, T_p]$= *Parent_Switching*(u_i, T_{ho})
4. if $P == 1$ // T_{ho} updated to T_p
5. $T_{ex} = T_p$
6. Calculate D_{ex}
7. return (P, T_{ex}, D_{ex});
8. else continue;
9. end for;
10. Identify node m with minimum lifetime in $C(T_{ho}, bn)$
11. if $h_m >= k$
12. return (P, T_{ho}, D_{ho})
13. else
14. *Extend_L_bn* $(m, T_{ho}, D_{ho}, G(V, E))$
15. end if

function *Tree_with_Y_leafnodes*
Input: $G(V, E)$
Output: W, T_l

1. $W = 0$;
2. Find set Y of low residual energy nodes in G such that for each node $y \in Y$, $\epsilon_y < \mathcal{E}_y$
3. Let $G' = (V', E')$ such that $V' = V - Y$ and $E' \in E$
4. Construct tree T' for G'
5. Update T' to T_l such that for each $y \in Y, C(T_l, y) = 0$
6. if T_l is constructed successfully
7. $W = 1$
8. else $T_l = 0$
9. end if
10. return (W, T_l)

function *Parent_Switching*
Input: u, T
Output: P, T_p

1. $P = 0$;
2. Let $VP(u) = \{v_1, v_2...v_{un}\}$ such that $L_{v_1} >= L_{v_2} >= ... >= L_{v_{un}}$
3. for each node $v_i \in VP(u)$ do
4. $Z_{v_i} = Z_{v_i} + Z_u$
5. Compute $P_{v_i}, \mathcal{E}_{v_i}, L_{v_i}$
6. if $L_{v_i} > L_{c_u}$
7. Add edge (u, v_i) to T
8. Remove edge (u, c_u) from T
9. Update $Z_{u_i} P_{u_i}, \mathcal{E}_{u_i}, L_{u_i}$ for nodes on path to s
10. $P = 1$;
11. return (P, T)
12. else continue
13. end for
14. return(P, T)

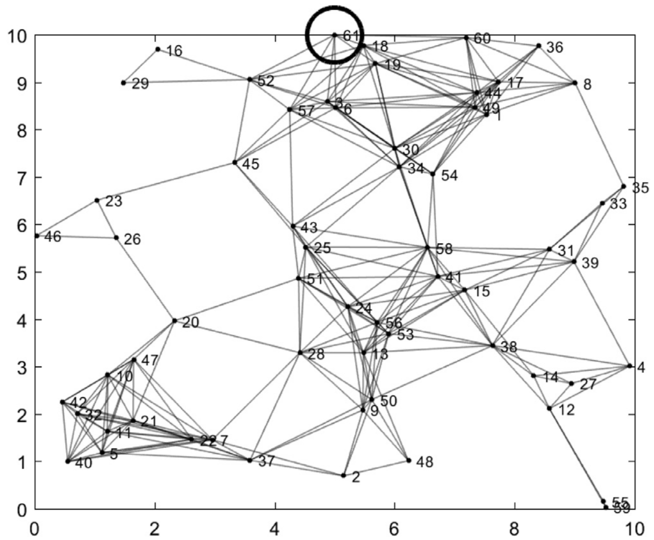

Fig. 3. An example connectivity graph $G = (V, E)$ representing initial random deployment of 60 nodes placed in a 10×10 area with transmission range $g = 2$. Encircled node 61 is the sink s.

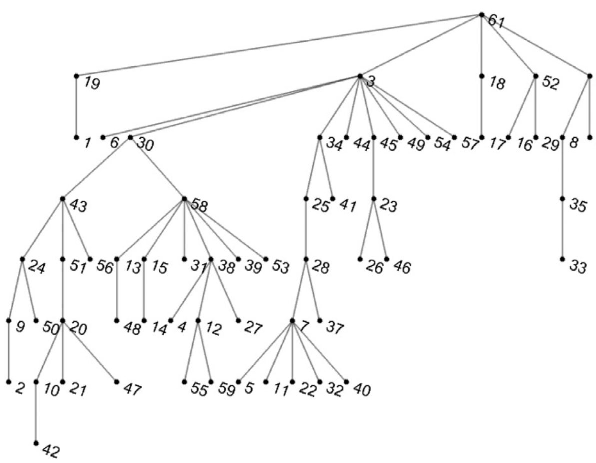

Fig. 4. SDLHO,$N = 60$, Step 1 :Initial DAT T_{isp}.

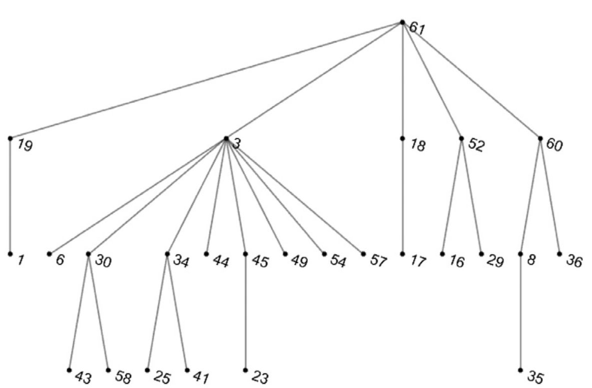

Fig. 5. SDLHO,$N = 60$, Step 2 : T a k-hop subtree of T_{isp}, $D = 12$.

4) *node*_3 and *node*_18 are identified as the bottleneck nodes and these nodes execute *Extend_L_bn* to get reconstructed DAT T_{ex}, Fig. 7. Fifth column in Table 1 represents $\mathcal{E}_u$ for T_{ex} and $D_{ex} = 32$ rounds.
5) Schedule U is updated to $U = \{T_1 = T_{ex}\}, \{D_1 = D_{ex} = 32\}$.
6) The set of low residual energy nodes $Y = \{node\_3, node\_8\}$ is identified and tree T_l (Fig. 8) is constructed by executing

Table 1
Energy *units* required at each node for DATs T, T_{ho}, T_{ex} for SDLHO is represented by columns $\mathcal{E}_u[T]$, $\mathcal{E}_u[T_{ho}]$ and $\mathcal{E}_u[T_{ex}]$, respectively. Initial energy at all nodes is $\epsilon = 2000$*units*.

I Node u	II δ_u	III $\mathcal{E}_u[T]$	IV $\mathcal{E}_u[T_{ho}]$	V $\mathcal{E}_u[T_{ex}]$
1	0	0	0	0
3	4	155	71	62
6	10	8	8	8
8	10	9	9	9
16	2	2	2	2
17	0	0	0	0
18	4	4	70	61
19	5	4	13	31
23	4	5	5	5
25	6	16	16	16
29	10	8	8	8
30	9	84	51	51
34	9	31	6	24
35	2	2	2	2
36	0	0	0	0
41	7	6	6	6
43	0	35	35	35
44	9	6	6	6
45	4	8	59	41
49	3	2	2	2
52	6	17	29	29
54	0	0	9	9
57	0	0	0	0
58	9	43	43	43
60	3	14	14	14
D		12	28	32

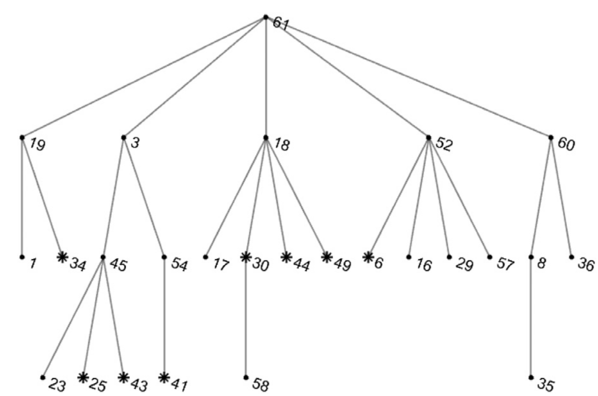

Fig. 6. SDLHO,$N = 60$, Step 3 : DAT T_{ho}. All * marked nodes 6,25,30,34,43,49 have switched to new parent nodes, $D_{ho} = 28$.

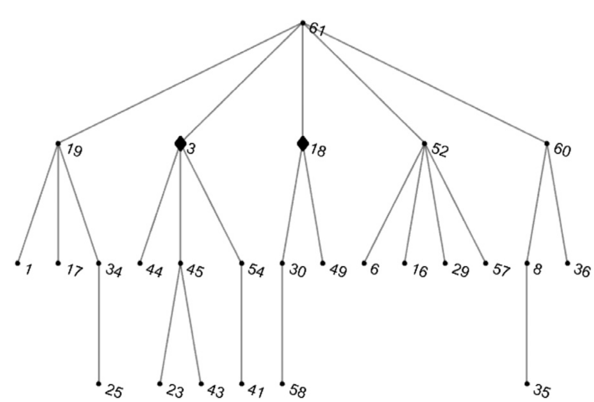

Fig. 7. SDLHO,$N = 60$, Step 4 : DAT T_{ex} with bottleneck nodes node_3 and node_18. $D_{ex} = 32$.

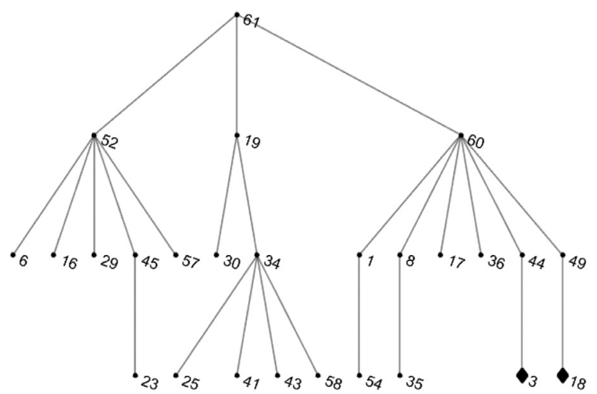

Fig. 8. SDLHO,$N = 60$, Step 6 : T_l with low residual energy nodes *node*_3 and *node*_18 as leaf nodes, $D_l = 4$.

Tree_with_Y_leafnodes. Leaf nodes *node*_3, *node*_8 consume less energy as they do not receive data from child nodes. Steps 3 to 6 of SDLHO are reiterated and T_l, D_l is added to schedule U.

Final schedule is represented by U={T_1, T_2}, {32, 4} and DATs are given in Fig. 7 and 8 respectively. The maximum lifetime D is given by 32+4=36 rounds. As seen from Table 1, the maximum energy required by a node is reduced by reconstructing the tree so that energy consumed by all nodes is balanced improving NL.

3.6. Complexity analysis for SDLHO

It is assumed that the network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ has N nodes. Let n be the number of nodes in $G(V, E)$ such that $|V| = |V_T| = n$. H is the maximum number of hops required to reach the sink in DAT $T(V_T, E_T)$ and Q is the maximum number of neighboring nodes of any node in G, $n > Q$.

Stepwise computation and communication complexity

1) The communication complexity of initial DAT construction is given by the number of single hop messages. Each node broadcasts a message to its maximum Q neighbors resulting in communication complexity of $O(N \cdot Q)$.
2) The sink constructs DAT $T(V_T, E_T)$ by using parent and child information of nodes at distance k from itself. This requires $|V_T| = n$ message transmissions to sink, resulting in communication complexity of $O(n)$
3) In *Local_Heuristics*, the for loop in line 1 is executed n times, once for each node. Inside for loop, *Parent_Switching* is executed. The computational complexity of *Parent_Switching* is determined by the number of valid parents $O(Q)$ of a node and the path length H. Hence, the complexity of *Parent_Switching* is $O(H \cdot Q)$ and the complexity of *Local_Heuristics* is $O(n \cdot H \cdot Q)$
4) *Extend_L_bn* is executed for each bottleneck node. Maximum number of bottleneck nodes is O(Q). Hence the computational complexity of this step is $O(Q) \times$ complexity of *Extend_L_bn*.
   In *Extend_L_bn*, the for loop in line 1 is executed $O(Q)$ times. Inside for loop, *Parent_Switching* is called. The recursive *Extend_L_bn* function executes at most H times. Thus, the computational complexity of *Extend_L_bn* is $O(H^2 \cdot Q^2)$ and the complexity of step 4 is $O(H^2 \cdot Q^3)$
5) Constant time $O(1)$ is required to add T_{ex} and D_{ex} to U. Evaluating remaining energy of all nodes has $O(n)$ complexity.
6) *Tree_with_Y_leafnodes* constructs T_l with complexity $O(n^2)$. It is observed that steps 3 to 6 are executed constant c number of times.

The total computational complexity of SDLHO is

$$(O(n \cdot H \cdot Q) + O(H^2 \cdot Q^3) + O(n) + O(n^2)) = O(n^2 + H^2 \cdot Q^3)$$

3.7. SDLHO Heuristics for imperfect links

SDLHO models perfect communication links with no associated uncertainty. However, in reality, environmental factors cause multipath propagation and distortion of signals and lead to radio link quality deterioration in WSNs [33,34]. A key technique for modeling imperfect links is based on using Link Quality Indicator (LQI) values offered by the corresponding radio chips [37].

In the proposed work, Scheduling DATs using Local Heuristics with Ordering based on Link Quality (SDLHO-LQ) algorithm is presented.

3.7.1. Algorithm SDLHO-LQ

SDLHO-LQ extends SDLHO Local Heuristics to assure best possible communication links between the nodes and suits practical scenarios. In SDLHO-LQ, each node u analyses its valid parent nodes $VP(u)$ to check if changing parent node c_u improves L_{c_u} while maintaining assured link quality. The *Parent_Switching* function for SDLHO-LQ selects a valid parent v_i with maximum link weight. This strategy assures that the nodes always communicate over the best possible link to address the challenges of uncertain communication links.

3.8. Algorithm SDLHR: Scheduling DATs using Local Heuristics with Randomization

The proposed SDLHR algorithm uses randomization to find a schedule of DATs. SDLHR employs Local Heuristics at each node. Each node u analyses its valid parent nodes $VP(u)$ to check if changing parent node c_u improves L_{c_u}. SDLHR differs from SDLHO in its parent switching mechanism.

In SDLHO, line 1–2 of *Parent_Switching* function select a valid parent v_i in decreasing order of its lifetime L_{v_i}. Ordering induces computational overhead at the node. Therefore with the objective of proposing Local Heuristics which is lightweight with less computational overhead, SDLHR is proposed. SDLHR uses randomization in *Parent_Switching* and the ordering complexity is eliminated.

4. Algorithm SDLHT: Scheduling DATs using Local Heuristics with Tree factor

The maximum number of rounds for which a given DAT T can be scheduled is its lifetime D. Whether a new DAT should be constructed after D rounds or before D rounds is a challenging problem. If the DAT is reconstructed after every DA round then the lifetime of the network can be improved. This motivates us to investigate a new basis for tree reconstruction termed as tree factor *tf*. In this section, tree factor is described and Scheduling DATs using Local Heuristics with Tree factor (SDLHT) algorithm is proposed.

4.1. Tree factor

Tree factor *tf* such that $tf \in Z^+$ controls the number of data collection rounds for a given tree. Let $U = \{T_1, T_2, \ldots T_t\}, \{D_1, D_2, .D_t\}$ represent a tree schedule with t trees. The parameter *tf* is application dependent and is used to control the number of data aggregation rounds for each T_i. Let D_i be the lifetime of T_i then after D_i rounds some nodes in the network cannot relay data and the tree has to be reconstructed. With *tf*, the tree is checked earlier for possible tree reconstruction when p percentage of nodes' energy is depleted. The value of p is determined by *tf* and is denoted as $p = 100/tf$. For example, for a given value of *tf*, if the current tree is T_i with lifetime D_i, then T_i is checked for reconstruction after d_i rounds, $d_i \le D_i$ such that

$$d_i = \lfloor D_i/tf \rfloor \tag{13}$$

In case of SDLHO and SDLHR $tf = 1$ and $d_i = D_i$.

4.2. SDLHT Algorithm

The proposed SDLHT employs hybrid DAM II (Ref. Section 2.2.3). After the initial DAT construction, the k-hop nodes reconstruct the tree using Local Heuristics and *tf* (Algorithm 2).

Algorithm 2 SDLHT.

Input: A randomly deployed WSN as graph $G = (V, E)$ with N sensor nodes and a sink node s. δ_u and ϵ_u for each node $u \in V$ representing data generated by the node and its initial energy. Tx, Rx is energy required for transmitting and receiving a packet respectively.tf is tree factor.

Output: A schedule $U = \{T_1, T_2, ...T_t\}, \{d_1, d_2, ..d_t\}$

Steps

1. Construct DAT in a distributed manner to get $T_{isp} = (V_{isp}, E_{isp})$ such that $V_{isp} = V, E_{isp} \subseteq E$. Each node computes $Z_u, P_u, \mathcal{E}_u, L_u$.
2. Nodes within k-hop distance in T_{isp} form T. Each node in T forwards the value of L_u on the path towards the sink and sink computes D.
3. T is reconstructed to T_{ho} by each node u executing *Parent_Switching* function. Each node then forwards L_u to sink.
4. Sink computes D_{ho} and d_{ho} using Eq.(10) and Eq. (13) respectively. Initially $i = 0$. Schedule U is updated in each iteration by adding T_{ho}, d_{ho} to U such that $i = i + 1, T_i = T_{ho}, d_i = d_{ho}$. Sink then communicates value of d_{ho} to all nodes in T_{ho}. Nodes in T_{ho} receive d_{ho} and update ϵ_u accordingly. if $\epsilon_u < \mathcal{E}_u$ then tree cannot be constructed and U is returned, otherwise steps 3 and 4 are repeated.

4.3. SDLHT example

A randomly deployed sensor network of 70 nodes with node 71 as sink node is shown in Fig. 9. Let $tf = 6$. A schedule of heuristic

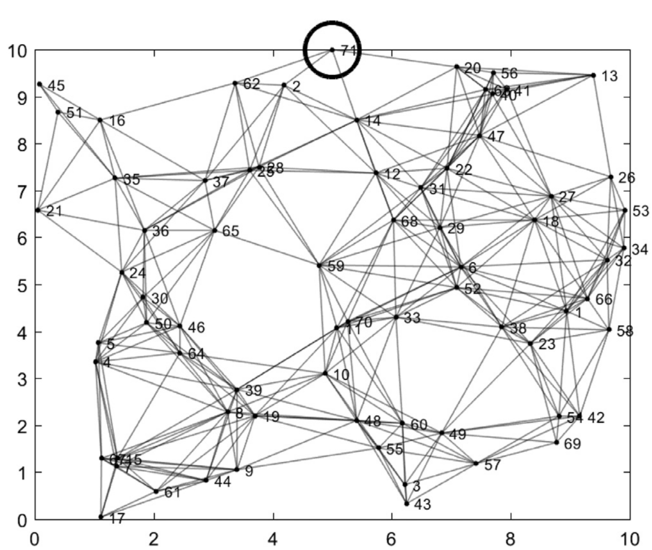

Fig. 9. G(V,E) representing initial random deployment of 70 nodes placed in a 10×10 area with transmission range $g = 2$. Encircled node 71 is the sink.

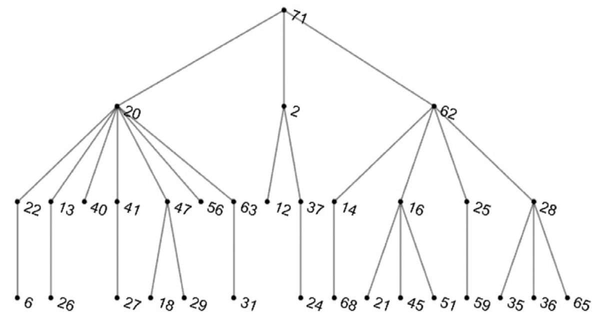

Fig. 10. SDLHT,$N = 70$, step 4, i=1 : T_1 of schedule U, $d_1 = 2$.

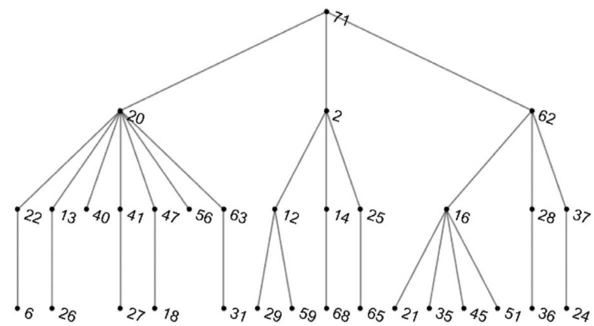

Fig. 11. SDLHT,$N = 70$, step 4, i=2 : T_2 of schedule U, $d_2 = 2$.

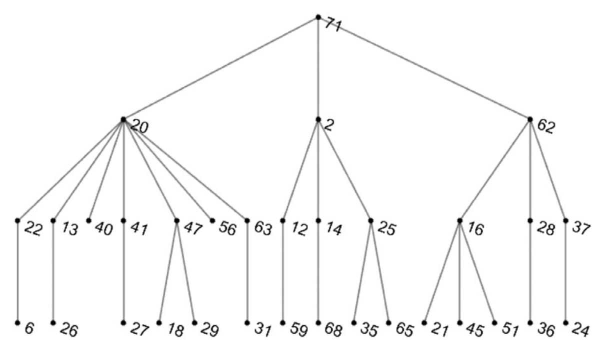

Fig. 12. SDLHT,$N = 70$, step 4, i=3 : T_3 of schedule U, $d_3 = 10$.

trees constructed with SDLHT is given in Figs. 10, 11 and 12. Final schedule is represented as $U = \{T_1, T_2, T_3\}, \{2, 2, 10\}$.

4.4. Complexity analysis of SDLHT

In SDLHT, the initial DAT is constructed in a distributed manner as in SDLHO and SDLHR algorithms. Hence the communication complexity is $O(N \cdot Q)$. In steps 2,3 and 4, n messages are transmitted adding communication complexity given by $O(n)$. Computational complexity is given by $O(n \cdot H \cdot Q)$ as each node executes *Parent_Switching* with computational complexity $O(H \cdot Q)$.

5. Performance evaluation

5.1. Simulation setup and parameters

In the simulation setup, WSNs are generated by deploying relay and source sensor nodes in the are of interest. To facilitate performance comparison, the simulation setup is similar to that of LTRBSA [22].

N sensors are randomly deployed where $|N|$ varies between 10–100 with step of 10 and 100–1000 nodes with step of 100. Deployment is in a 10×10 sq.units field with transmission range at each node $g = 2$. The number of relay nodes $|R| = 50$. Initial energy, $\epsilon = 10^5$ units, energy required to transmit a packet is $Tx = 2$ and energy required to receive a packet is $Rx = 1$. The data bits generated by source nodes is randomly selected from the interval 1 to B where $B = 10$ units. Data aggregation ratio $\alpha = 3$ (Eq. (4)). Value of k for k-hop information is set to $k = 3$. Perfect and imperfect links are considered (Refer Section 2.1). Thirty networks are generated for each value of $|N|$ and the values of NL are observed. Mean and standard deviation values of energy consumed by all nodes in a DA round are also observed.

5.2. Observations for network lifetime

Network lifetime values for number of sensors ranging from 10 to 100 and 100–1000 are given in Figs. 13 and 14, respectively. For $|N|$ between 10–100 and $|R|$=0, the NL values for SDLHO, SDLHR and LTRBSA [22] algorithms are similar. However, when $|N|$ ranges between 100–1000 and $|R|$=50, NL values reported by SDLHO and SDLHR show significant improvement compared to LTRBSA [22].

Simulations were also performed to compare SDLHO and SDLHR algorithms with state-of-art. Results for NL values of SDLHO and SDLHR are compared with RaSMaLai [50] and MITT [51]. To facilitate performance comparison, the simulation setup in this case is similar to RaSMaLai and MITT. Deployment is in a 100×100 sq. units field with transmission range at each node $g = 25$. The sink is placed at the centre of the deployment area. The number of sensors are between 100–400 with a step of 100. Simulation results given in Fig. 17 show that SDLHO improves NL as compared to state-of-art.

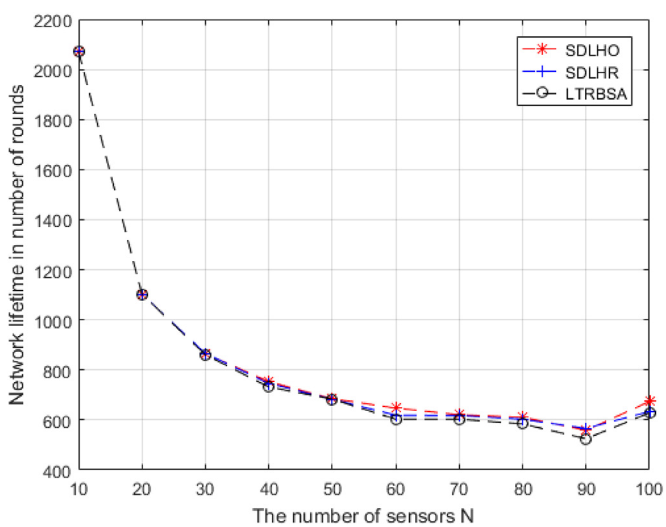

Fig. 13. The lifetime of networks with number of sensors ranging from 10 to 100.

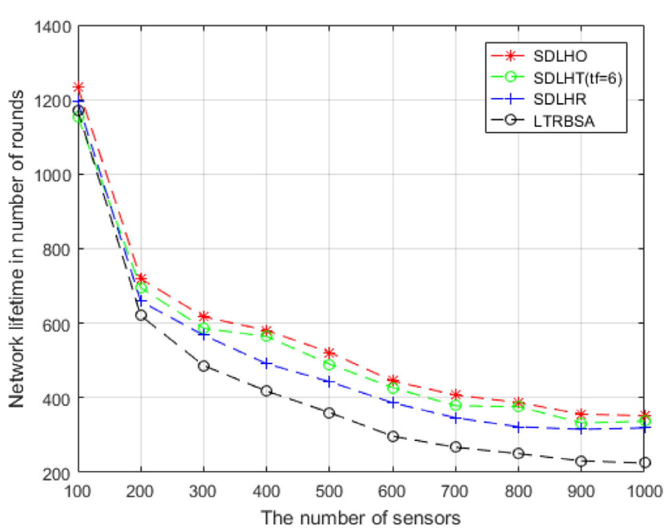

Fig. 14. The lifetime of networks with number of sensors ranging from 100 to 1000.

5.3. Observations for energy depleted in a DA round

The mean and standard deviation of energy depleted by the nodes per DA round is shown in Fig. 15 and Fig. 16, respectively. It is observed that the mean energy depleted is similar for SDLHO, SDLHR and LTRBSA [22]. However SDLHO has lower values for standard deviation compared to SDLHR. Standard deviation values for LTRBSA [22] are highest. Lower standard deviation values signify balanced energy consumption that leads to improved network lifetime in SDLHO.

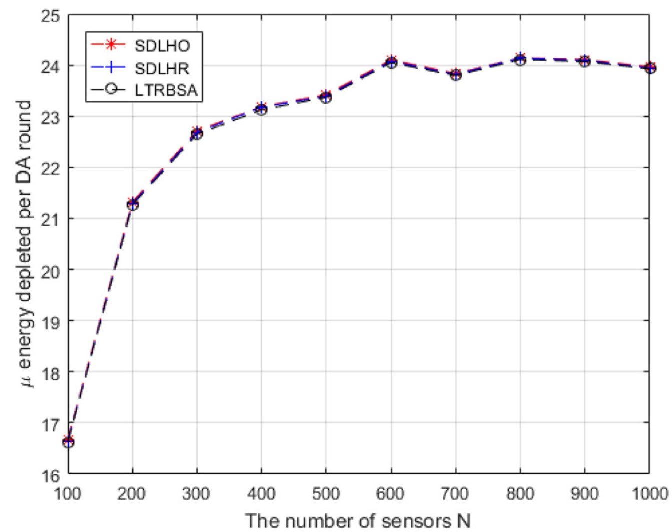

Fig. 15. Mean energy depleted in each round for networks with number of sensors ranging from 100 to 1000.

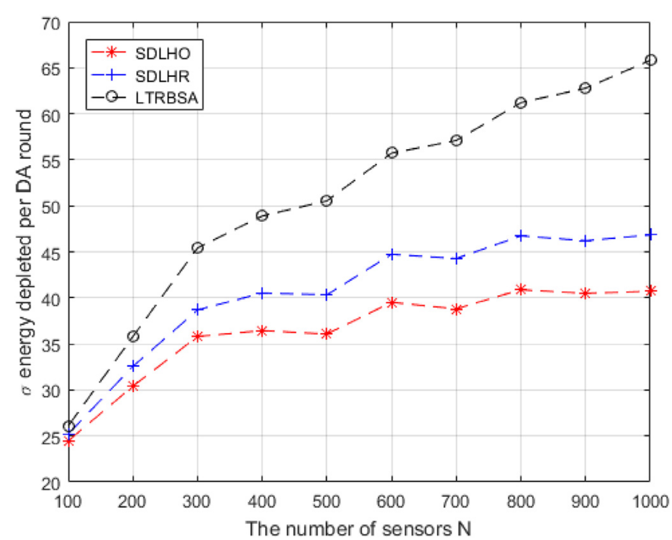

Fig. 16. Standard deviation of energy depleted for networks with number of sensors ranging from 100 to 1000.

Table 2
Comparison of algorithms LTRBSA [22], SDLHO, SDLHR and SDLHT.

Parameters	LTRBSA [22]	SDLHO	SDLHR	SDLHT
Local Heuristics	Selects neighboring node ordered by residual energy	Selects neighboring node with maximum residual energy	Selects neighboring node randomly	Selects neighboring node with maximum residual energy
DAT reconstruction condition	$\epsilon_u < \mathcal{E}_u$ for some node u in the network	$\epsilon_u < \mathcal{E}_u$ for some node u in the network	$\epsilon_u < \mathcal{E}_u$ for some node u in the network.	Depends on tree factor *tf* and ϵ_u.
DAM model employed.	–	Hybrid DAM I	Hybrid DAM I	Hybrid DAM II
Computational complexity.	$O(n^2 + n \cdot H^2 \cdot Q^2)$	$O(n^2 + H^2 \cdot Q^3)$	$O(n^2 + H^2 \cdot Q^3)$	$O(n \cdot H \cdot Q)$
NL in terms of DA rounds, $\epsilon_u = 10^5$	1171 rounds for N=100 360 rounds for N=500 224 rounds for N=1000	1234 rounds for N=100 521 rounds for N=500 352 rounds for N=1000	1193 rounds for N=100 443 rounds for N=500 319 rounds for N=1000	1154 rounds for N=100 490 rounds for N=500 337 rounds for N=1000
Percentage improvement in NL	Compared to SPTBSA: 32.4% for N=200; 44.3% for N=500; 50.1% for N=1000 (Ref. Fig 5 in [22])	Compared to LTRBSA: 15.9% for N=200; 44.8% for N=500; 57.1% for N=1000	Compared to LTRBSA: 6.21% for N=200; 23.0% for N=500; 42.0% for N=1000.	Compared to LTRBSA: 12.1% for N=200; 36.0% for N=500; 50.0% for N=1000

5.4. Observations for scalability

NL values of the proposed algorithms are compared in Fig. 14. LTRBSA has better NL values compared to SPTBSA [22]. The proposed algorithms are compared with LTRBSA [22]. Fig. 14 and Table 2 present experimental results and show that the SDLHO, SDLHT and SDLHR are highly scalable and achieve more than 50% performance improvement as the number of nodes in the network increases.

5.5. Observations with imperfect links

NL values for sensor networks with imperfect links, using SDLHO-LQ, is given in Fig. 18. Each data point in the simulations was averaged over thirty networks and scaled between 0 to 1. The results show the impact of link quality on the observed NL values. SDLHO-LQ trades NL for improved link quality and suits realistic deployment scenarios.

The link weights can be modified to suit the deployment environment and the devices used [38,52,53].

5.6. Observations for tree factor *tf*

Simulation results presented in Fig. 14 have *tf* set to 6 for SDLHT algorithm. This value is obtained empirically. Fig. 19 shows NL values for *tf* ranging from 2 to 14 and $|N|$ between 100–1000 nodes. It is observed that for $tf = 2$, NL is less as compared to higher values of *tf*. Increasing *tf* increases the frequency at which the DAT is checked for possible reconstruction and results in improved chances for nodes to find parent nodes with better lifetime and effectively increases the NL. However increasing *tf* also increases tree reconstruction overhead. Fig. 19 shows that increasing *tf* above 6 does not achieve significant NL improvement. Hence *tf* is set to 6 for SDLHT.

5.7. Comparison of algorithms

Comparison of the proposed algorithms and state of art is described in Table 2. It is seen that Local Heuristics based on residual energy of neighboring nodes can be tuned to application needs to achieve desired NL improvement. In SDLHO and SDLHR, DAT is reconstructed if residual energy at a node is insufficient. In SDLHT, DAT is reconstructed if residual energy is less than a certain threshold value. This value is controlled by using tree factor. SDLHT employs hybrid DAM II model. Hybrid DAM II is distributed and overcomes the challenges of Hybrid DAM I. SDLHT is computationally most efficient as described in Table 2. Improvement in NL values of the proposed algorithms as the network size grows make them suitable for applications that require scalability.

To tackle uncertainty in communication in real world sensor networks, the heuristics is adapted so that it caters to realistic scenarios. SDLHO heuristics for imperfect links is extended to accomodate good quality links in SDLHO-LQ. The capability of SDLHO-LQ

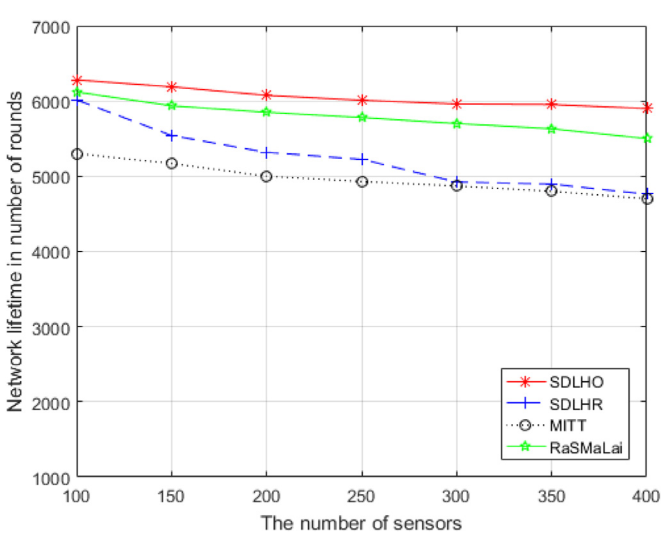

Fig. 17. The lifetime of networks with number of sensors ranging from 100 to 400.

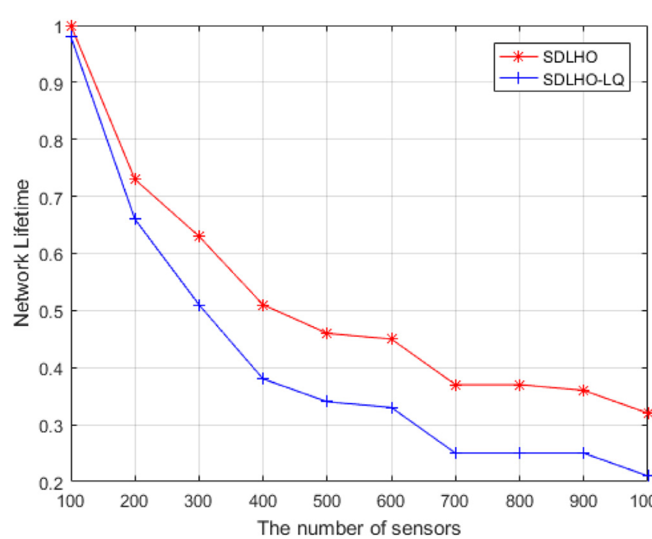

Fig. 18. The lifetime of networks with imperfect links for number of sensors ranging from 100 to 1000.

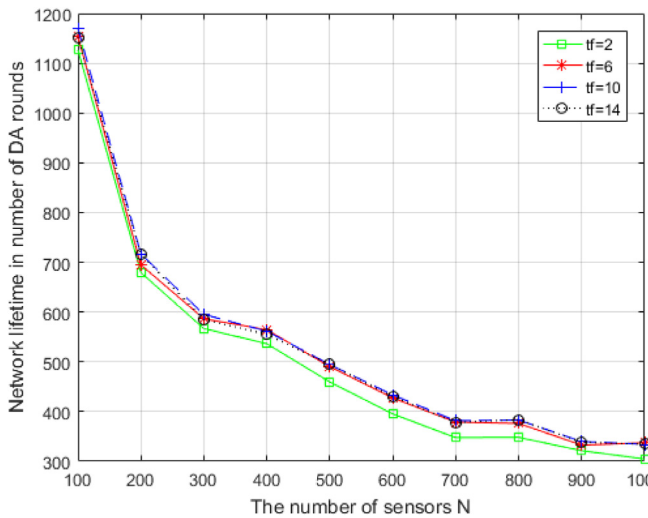

Fig. 19. The lifetime of networks with SDLHT that have tree factor ranging between 2 to 14 and number of sensors ranging from 100 to 1000.

heuristics is suitable in practical scenarios where shorter NL can be traded for achieving assured communication over good quality links.

6. Conclusion

In WSNs, the structure of the DAT and the number of packet transmissions influences the energy requirements and affects NL values. Data is gathered in an energy efficient way by constructing DATs.

In this paper, the problem of enhancing Network Lifetime (NL) using hybrid DAT construction methods is investigated and three algorithms for Scheduling DATs using Local Heuristics with Ordering (SDLHO), with Randomization (SDLHR) and with Tree factor (SDLHT) techniques are proposed.

SDLHO schedules a sequence of DATs and employs hybrid DAM I (Ref. Section 2.2.3). Local heuristics implemented in SDLHO is based on the residual energy of neighboring nodes and requires ordering at node level. To eliminate this computational overhead, a lightweight, randomization based Local Heuristics for DAT construction is proposed and implemented in SDLHR Scheduling DATs using Local Heuristics with Randomization algorithm. Both, SDLHO and SDLHR employ hybrid DAM I in which the sink requires partial knowledge of the network.

To address the concerns regarding behavior of wireless links in practical situations, the heuristics of SDLHO is modified in the proposed Scheduling DATs using Local Heuristics with Ordering based on Link quality (SDLHO-LQ) algorithm. Results demonstrate the applicability of the proposed SDLHO-LQ to suit the uncertainty in real world application scenarios.

To overcome requirements of awareness at sink node, a new basis for tree reconstruction termed tree factor is investigated and proposed. Tree factor based technique constructs the DAT in a distributed manner as implemented and demonstrated by Scheduling DATs using Local Heuristics with Tree factor (SDLHT) algorithm.

Rigorous simulation results demonstrate the efficacy of the proposed algorithms. SDLHO, SDLHR and SDLHT are scalable and improve NL as compared to existing state of art by $5-50\%$ for the random deployment of $100-1000$ nodes, respectively in common platform settings. (Ref. Table 2) Their ability to scaleup to suit deployment of applications in harsh regions is promising and suits distributed environments.

Enhancing network lifetime in dynamic environments with mobile nodes impose challenges for future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

References

[1] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, Tag: a tiny aggregation service for ad-hoc sensor networks, SIGOPS Oper. Syst. Rev. 36 (SI) (2002) 131–146, doi:10.1145/844128.844142.
[2] T.-W. Kuo, K.C.-J. Lin, M.-J. Tsai, On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: np-completeness and approximation algorithms, IEEE Trans. Comput. 65 (10) (2016) 3109–3121.
[3] H.Ö. Tan, I. KörpeoÇ§lu, Power efficient data gathering and aggregation in wireless sensor networks, ACM SIGMOD Rec. 32 (4) (2003) 66–71.
[4] Y. Wu, Z. Mao, S. Fahmy, N.B. Shroff, Constructing maximum-lifetime data–gathering forests in sensor networks, IEEE/ACM Trans. Netw. 18 (5) (2010) 1571–1584.
[5] M. Shan, G. Chen, D. Luo, X. Zhu, X. Wu, Building maximum lifetime shortest path data aggregation trees in wireless sensor networks, ACM Trans. Sensor Netw. (TOSN) 11 (1) (2014) 11.
[6] F. Zhou, Z. Chen, S. Guo, J. Li, Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs, IEEE Sens. J. 16 (22) (2016) 8167–8177.
[7] Y. Lu, I.-S. Comsa, P. Kuonen, B. Hirsbrunner, Adaptive data aggregation with probabilistic routing in wireless sensor networks, Wirel. Netw. 22 (8) (2016) 2485–2499.
[8] J. He, S. Ji, Y. Pan, Y. Li, Constructing load-balanced data aggregation trees in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 25 (7) (2014) 1681–1690.
[9] M. Ding, X. Cheng, G. Xue, Aggregation tree construction in sensor networks, in: Proceedings of the IEEE 58th Vehicular Technology Conference, VTC, 4, IEEE, 2003, pp. 2168–2172.
[10] M. Lee, V.W. Wong, An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks, in: Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and signal Processing, IEEE, 2005, pp. 300–303.
[11] L. Krishnamachari, D. Estrin, S. Wicker, The impact of data aggregation in wireless sensor networks, in: Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, IEEE, 2002, pp. 575–578.
[12] H. Luo, Y. Liu, S.K. Das, Routing correlated data in wireless sensor networks: a survey, IEEE Netw. 21 (6) (2007).
[13] Y. Zhu, K. Sundaresan, R. Sivakumar, Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks, in: Proceedings of the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE, 2005, pp. 328–339.
[14] R. Cristescu, B. Beferull-Lozano, M. Vetterli, R. Wattenhofer, Network correlated data gathering with explicit communication: np-completeness and algorithms, IEEE/ACM Trans. Netw. 14 (1) (2006) 41–54.
[15] H. Luo, J. Luo, Y. Liu, S.K. Das, Adaptive data fusion for energy efficient routing in wireless sensor networks, IEEE Trans. Comput. 55 (10) (2006) 1286–1299.
[16] B.-H. Liu, J.-Y. Jhang, Efficient distributed data scheduling algorithm for data aggregation in wireless sensor networks, Comput. Netw. 65 (2014) 73–83, doi:10.1016/j.comnet.2014.03.003.
[17] C. Buragohain, D. Agrawal, S. Suri, Power aware routing for sensor databases, in: Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 3, IEEE, 2005, pp. 1747–1757.
[18] K. Dasgupta, K. Kalpakis, P. Namjoshi, An efficient clustering-based heuristic for data gathering and aggregation in sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking, 3, IEEE, 2003, pp. 1948–1953.
[19] Y. Xue, Y. Cui, K. Nahrstedt, Maximizing lifetime for data aggregation in wireless sensor networks, Mobile Netw. Appl. 10 (6) (2005) 853–864.
[20] H.-C. Lin, F.-J. Li, K.-Y. Wang, Constructing maximum-lifetime data gathering trees in sensor networks with data aggregation, in: Proceedings of the IEEE International Conference on Communications (ICC), IEEE, 2010, pp. 1–6.
[21] H. Matsuura, Maximizing lifetime of multiple data aggregation trees in wireless sensor networks, in: Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE, 2016, pp. 605–611.
[22] N.-T. Nguyen, B.-H. Liu, V.-T. Pham, Y.-S. Luo, On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data aggregation trees, Comput. Netw. 105 (C) (2016) 99–110, doi:10.1016/j.comnet.2016.05.022.
[23] Y. Sankarasubramaniam, I.F. Akyildiz, S. McLaughlin, Energy efficiency based packet size optimization in wireless sensor networks, in: Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications, IEEE, 2003, pp. 1–8.

[24] C. Majumdar, D. Lee, A.A. Patel, S. Merchant, U. Desai, Packet size optimization for cognitive radio sensor networks aided internet of things, IEEE Access 5 (2017) 6325–6344.
[25] P. Kale, M.J. Nene, Path reestablishment in wireless sensor networks, in: Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 1659–1663, doi:10.1109/WiSPNET.2017.8300043.
[26] S. Xie, Y. Wang, Construction of tree network with limited delivery latency in homogeneous wireless sensor networks, Wirel. Pers. Commun. 78 (1) (2014) 231–246, doi:10.1007/s11277-014-1748-5.
[27] Y.E. Ahmed, K.H. Adjallah, R. Stock, I. Kacem, S.F. Babiker, NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring, Comput. Ind. Eng. 115 (2018) 17–25, doi:10.1016/j.cie.2017.09.049.
[28] R.M. Al-Kiyumi, C.H. Foh, S. Vural, P. Chatzimisios, R. Tafazolli, Fuzzy logic-based routing algorithm for lifetime enhancement in heterogeneous wireless sensor networks, IEEE Trans. Green Commun. Netw. 2 (2) (2018) 517–532, doi:10.1109/TGCN.2018.2799868.
[29] H.C. Lin, W.Y. Chen, An approximation algorithm for the maximum-lifetime data aggregation tree problem in wireless sensor networks, IEEE Trans. Wireless Commun. 16 (6) (2017) 3787–3798, doi:10.1109/TWC.2017.2688442.
[30] H. Yetgin, K.T.K. Cheung, M. El-Hajjar, L.H. Hanzo, A survey of network lifetime maximization techniques in wireless sensor networks, IEEE Commun. Surv. Tutor. 19 (2) (2017) 828–854, doi:10.1109/COMST.2017.2650979.
[31] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, 2003, pp. 14–27.
[32] G. Zhou, T. He, S. Krishnamurthy, J.A. Stankovic, Impact of radio irregularity on wireless sensor networks, in: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, in: MobiSys '04, ACM, New York, NY, USA, 2004, pp. 125–138, doi:10.1145/990064.990081.
[33] C.A. Boano, N. Tsiftes, T. Voigt, J. Brown, U. Roedig, The impact of temperature on outdoor industrial sensornet applications, IEEE Trans. Ind. Inf. 6 (3) (2010) 451–459.
[34] K. Bannister, G. Giorgetti, S. Gupta, Wireless sensor networking for hot applications: effects of temperature on signal strength, data collection and localization, in: Proceedings of the 5th Workshop on Embedded Networked Sensors (HotEmNets 08), Citeseer, 2008.
[35] T. Yang, Q. Yang, L. Cheng, Experimental study: a LQI-based ranging technique in ZigBee sensor networks, Int. J. Sens. Netw. 19 (2) (2015) 130–138.
[36] N. Baccour, A. Koubâa, L. Mottola, M.A. Zúñiga, H. Youssef, C.A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: a survey, ACM Trans. Sens. Netw. (TOSN) 8 (4) (2012) 34.
[37] C. Gomez, A. Boix, J. Paradells, Impact of LQI-based routing metrics on the performance of a one-to-one routing protocol for ieee 802.15. 4 multihop networks, EURASIP J. Wirel. Commun. Netw. 2010 (1) (2010) 205407.
[38] A. Bildea, O. Alphand, F. Rousseau, A. Duda, Link quality metrics in large scale indoor wireless sensor networks, in: Proceedings of the IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), IEEE, 2013, pp. 1888–1892.
[39] U. Raza, A. Camerra, A.L. Murphy, T. Palpanas, G.P. Picco, Practical data prediction for real-world wireless sensor networks, IEEE Trans. Knowl. Data Eng. 27 (8) (2015) 2231–2244.
[40] M. Zuniga, B. Krishnamachari, Analyzing the transitional region in low power wireless links, in: Proceedings of the SECON, 4, 2004, pp. 517–526.
[41] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, Understanding the causes of packet delivery success and failure in dense wireless sensor networks, in: Proceedings of the 4th international Conference on Embedded Networked Sensor Systems, ACM, 2006, pp. 419–420.
[42] T. Liu, A.E. Cerpa, Data-driven link quality prediction using link features, ACM Trans. Sens. Netw. (TOSN) 10 (2) (2014) 37.
[43] J. He, S. Ji, R. Beyah, Y. Xie, Y. Li, Constructing load-balanced virtual backbones in probabilistic wireless sensor networks via multi-objective genetic algorithm, Trans. Emerg. Telecommun. Technol. 26 (2) (2015) 147–163.
[44] Y. Yao, Q. Cao, A.V. Vasilakos, Edal: an energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks, IEEE/ACM Trans. Netw. (TON) 23 (3) (2015) 810–823.
[45] C. Liu, G. Cao, Distributed monitoring and aggregation in wireless sensor networks, in: Proceedings of the INFOCOM, IEEE, 2010, pp. 1–9.
[46] S. Nazemi, K.K. Leung, A. Swami, Optimization framework with reduced complexity for sensor networks with in-network processing, in: Proceedings of the IEEE Wireless Communications and Networking Conference, IEEE, 2016, pp. 1–6.
[47] Y. Jin, S. Gormus, P. Kulkarni, M. Sooriyabandara, Content centric routing in IoT networks and its integration in RPL, Comput. Commun. 89 (C) (2016) 87–104, doi:10.1016/j.comcom.2016.03.005.
[48] V. Raghunathan, C. Schurgers, S. Park, M.B. Srivastava, Energy-aware wireless microsensor networks, IEEE Signal Process. Mag. 19 (2) (2002) 40–50.
[49] M.G. Calle Torres, Energy consumption in wireless sensor networks using GSP, University of Pittsburgh, 2006 Ph.D. thesis.
[50] S.K.A. Imon, A. Khan, M. Di Francesco, S.K. Das, Energy-efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks, IEEE/ACM Trans. Netw. (TON) 23 (5) (2015) 1401–1415.
[51] J. Liang, T. Li, A maximum lifetime algorithm for data gathering without aggregation in wireless sensor networks, Appl. Math. Inf. Sci. 7 (5) (2013) 1705.
[52] G. Zhou, T. He, S. Krishnamurthy, J.A. Stankovic, Models and solutions for radio irregularity in wireless sensor networks, ACM Trans. Sens. Netw. (TOSN) 2 (2) (2006) 221–262.
[53] J. Wen, Z. Ansar, W. Dargie, A link quality estimation model for energy-efficient wireless sensor networks, in: Proceedings of the IEEE International Conference on Communications (ICC), IEEE, 2015, pp. 6694–6700.

Preeti A. Kale received her Masters in computer Engineering from University of Pune, India. She is currently pursuing her Ph.D in computer science and engineering from Defence Institute of Advanced Technology (DIAT), Pune; a Defence Research and Development Organization (DRDO) establishment under Ministry of Defence, India. Her areas of interest are Wireless Sensor Networks and Analysis of Algorithms.

Manisha J. Nene is a faculty with the Department of Computer Engineering and Applied Mathematics, Defence Institute of Advanced Technology (DIAT), Pune; a Defence Research and Development Organization (DRDO) establishment under Ministry of Defence, India. She has 18 plus years of training and research experience. She is a recepient of national level awards for her contributions in ICT for societal development, training and research. Her areas of interest are Cyber Physical Systems, Adhoc and Self Organising Networks, Analysis of Algorithms, High Performance Computing, Data Analytics, Modelling and Simulation.