Computer Networks 53 (2009) 945–960

Contents lists available at ScienceDirect

Computer Networks

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

Data aggregation and routing in Wireless Sensor Networks: Optimal and heuristic algorithms

Jamal N. Al-Karaki, Raza Ul-Mustafa, Ahmed E. Kamal *

Iowa State University, Department of Electrical and Computer Engineering, Ames, IA 50011, United States

ARTICLE INFO

Article history:
Received 27 May 2008
Received in revised form 16 September 2008
Accepted 1 December 2008
Available online 11 December 2008

Responsible Editor: E. Ekici

Keywords:
Wireless Sensor Networks
Data aggregation and routing
Exact and heuristic solutions
Hierarchical
Clustering

ABSTRACT

A fundamental challenge in the design of Wireless Sensor Networks (WSNs) is to maximize their lifetimes especially when they have a limited and non-replenishable energy supply. To extend the network lifetime, power management and energy-efficient communication techniques at all layers become necessary. In this paper, we present solutions for the data gathering and routing problem with in-network aggregation in WSNs. Our objective is to maximize the network lifetime by utilizing data aggregation and in-network processing techniques. We particularly focus on the *joint* problem of optimal data routing with data aggregation en route such that the above mentioned objective is achieved. We present Grid-based Routing and Aggregator Selection Scheme (GRASS), a scheme for WSNs that can achieve low energy dissipation and low latency without sacrificing quality. GRASS embodies optimal (exact) as well as heuristic approaches to find the minimum number of aggregation points while routing data to the Base-Station (BS) such that the network lifetime is maximized. Our results show that, when compared to other schemes, GRASS improves system lifetime with acceptable levels of latency in data aggregation and without sacrificing data quality.

1. Introduction

Wireless Sensor Networks (WSNs) is a class of wireless ad hoc networks in which sensor nodes collect, process, and communicate data acquired from the physical environment to an external Base-Station (BS) [1]. Future WSNs are envisioned to revolutionize a maintenance free and fault-tolerant platform for collecting and processing information in diverse environments. A major technical challenge for WSNs, however, lies in the node energy constraint and its limited computing resources, which may pose a fundamental limit on the network lifetime [1]. Therefore, innovative techniques to eliminate energy inefficiencies that would otherwise shorten the lifetime of the network are highly needed.

In many applications of WSNs (e.g., military battlefields, target field imaging, intrusion detection, surveillance, and inventory control), data collected by many sensors is based on common phenomena, and hence there is a high probability that this data has some redundancy (or correlation). Due to the correlation present in the sensors' readings, it is expected that communication approaches that take into account this correlation, e.g., data aggregation and in-network processing, will outperform traditional approaches. The main idea of the data aggregation and in-network processing approaches is to combine the data arriving from different sources (sensor nodes) at certain aggregation points (or simply aggregators) en route, eliminate redundancies by performing simple processing at the aggregation points, and minimize the total amount of data transmission before forwarding data to the external BS. Removing redundancies results in transmitting fewer number of bits, and hence reduces energy consumption and increases the sensor nodes' lifetimes. A number of studies that compared aggregation scheme, e.g., [3–5,23]

* Corresponding author. Tel.: +1 515 294 3580; fax: +1 515 294 1152.
E-mail addresses: kamal@iastate.edu, aekamal@hotmail.com (A.E. Kamal).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.12.001

concluded that enhanced network throughput and more potential energy savings are highly possible using data aggregation and in-network processing in WSNs.

Aside from the task of efficient design of data aggregation algorithms, the task of finding and maintaining routes in WSNs is also nontrivial [2], especially when it includes the selection of aggregation points and routing through those points. Many routing and data dissemination with aggregation protocols have been proposed for WSNs (a comprehensive survey of the routing techniques in WSNs can be found in [2]). In [3], Intanagonwiwat et al. proposed a popular data aggregation paradigm for WSNs, called Directed Diffusion (DD) where aggregation is used to reduce communication costs. In [5], Heinzelman et al. introduced a hierarchical clustering algorithm for WSNs, called Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a cluster-based protocol where ClusterHead (CH) nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the BS. Following these protocols, many other studies focused on the routing problem [6–21]. Among these, [17] introduced a linear programming formulation to solve the optimal routing problem in WSNs. The objective was to maximize the network life time, and the life time was defined as the network operational time until one of the nodes fails. Necessary, but insufficient, conditions for the existence of a solution under arbitrary traffic generation processes were introduced. A heuristic solution was also introduced. However, no aggregation was assumed. In [18], aggregation was taken into account, but only full aggregation was considered. That is, regardless of the number of packets to be aggregated, a single packet will always be produced. This simplifies the problem of aggregation significantly. A special case of partial aggregation, in which the aggregated data size is equal to the minimum of the original data, and a maximum size, was considered in [19]. A heuristic was introduced for this case. In [20], heuristics as well as an approximation algorithm, were introduced to select the minimum number of sensors which would form the minimum connected correlation dominating set. Such a set is used to infer readings from other sensors (with an acceptable error), and also forward data towards the data collection center. The network lifetime was not explicitly addressed in this paper. The authors in [32] proposed a Voronoi detection method that utilizes distributed Voronoi diagram and genetic algorithms to gather data in WSNs, while [33] considers maximizing large scale Wireless Sensor Networks lifetime under energy constraint via joint relay deployment and adaptation. In [36], several lossless aggregate repacking algorithms for cluster-model Wireless Sensor Networks were presented. In [37], an energy-efficient data gathering scheme that prolongs the lifetime of battery-powered sensor nodes is considered. The proposed scheme constructs and maintains a spanning tree that is based on breadth-first search and has more leaf nodes in network. Kalpakis et al. [29] presented algorithms for the Maximum Lifetime Data Aggregation (MLDA) problem with the objective of maximizing the system lifetime. A near optimal algorithm for solving the MLDA problem was proposed in [29]. Since this algorithm is computationally expensive for large sensor networks, authors in [29] presented a clustering based heuristics approaches (CMLDA) for MLDA in large scale WSNs. Experimental results of MLDA demonstrated enhanced system lifetime of WSNs. Other related studies also appeared [28,30,31,34]. A recent survey of strategies and protocols for routing correlated data can be found in [35]. To the best of our knowledge, the joint optimization of the routing and data aggregation functions were not carried out in the literature, especially for the general case of partial aggregation.

In this paper, we present a novel data aggregation and routing scheme, called Grid-based Routing and Aggregator Selection Protocol (GRASS). GRASS embodies optimal (exact) as well as heuristic approaches to find the minimum number of aggregation points while routing data to the BS such that the network lifetime is maximized. That is, GRASS **jointly** addresses the issues of the selection of data aggregation points, and the optimal routing of data from sensors to aggregation points, as well as the routing of the aggregated data to the BS. While solving these two problems separately may simplify the problem, the solution may be far from optimal. Therefore, our proposed solution treats the two problems jointly in order to reach an optimal solution. Since this joint problem is not trivial, we adopt a hierarchical structure in which each group of sensor nodes elect a cluster head which is responsible for: (1) collecting their sensed data, (2) performing a first level aggregation, and then (3) routing this data to the next aggregator on its way to the BS. This first level of aggregation achieves two benefits. First, it offers the greatest performance benefits in this environment since nodes in a cluster are most likely to generate correlated data, and then it simplifies the routing function since only the cluster head will be in charge of this functionality. Hence, the hierarchical structure facilitates digests of sensor data. Indeed, this is a key issue in the design of GRASS. In GRASS, correlation means that sensors' readings overlap statistically as they monitor the same event. This overlap will be captured in our proposed solutions using aggregation overlap factor. The factor represents linear as well as non-linear relations among the gathered data. We propose to solve the joint aggregator selection and routing problems in a powerful node, such as the BS, and then dispatch the results to the sensor nodes. Hence, an optimal solution that is obtained by the BS will result in an optimal routing and aggregation strategy.

The rest of this paper is organized as follows: the problem description and system model are presented in Section 2. Section 3 presents exact algorithms to solve the problem, and using two definitions of network lifetime. Section 4 presents several approximate algorithms for the problem under consideration. Section 5 presents analysis of energy-delay tradeoffs due to our aggregation scheme. The performance evaluation of the proposed scheme is presented in Section 6. We conclude with final remarks in Section 7.

2. The problem description and system model

We consider a network of fixed, homogeneous, and energy-constrained sensor nodes that are randomly deployed in a sensor field (bounded region). Each sensor acquires measurements which are typically correlated with other sensors in its vicinity, and these measurements are to be gathered and sent to the BS for evaluation or decision taking purposes. We assume *periodic sensing* with the same period for all sensors. We also assume that contention between sensors is solved by the MAC layer.[1] We assume that the information collected by various sensors may be correlated, redundant, and/or of different qualities. Since data correlation in WSNs is strongest among data signals coming from nodes that are close to each other, we believe that the use of a clustering infrastructure will allow nodes that are close to each other to share data before sending it to the BS. Hence, the ideas of fixed cluster-based routing [27] together with application-specific data aggregation techniques will be used in GRASS to achieve significant energy savings. The virtual topology used in [27], namely VGA, has been proved to achieve significant reduction in control overhead and achieve more energy savings.

The concept of virtual topology presented in [27] has ramifications on the problem addressed in this paper. Therefore, we leverage this concept to perform energy efficient routing in WSNs. The essence of the clustering scheme presented in [27] is to create a fixed rectilinear virtual topology, called Virtual Grid Architecture (VGA), on top of the physical topology. VGA consists of a set of nodes, namely, ClusterHeads (CHs), that are elected periodically based on an eligibility criterion, which takes into account many changing parameters in the network. Each CH is elected as such inside a zone where we have divided the network area into fixed and square zones as shown in Fig. 1. The set of CHs form a fixed rectilinear virtual graph G. New CHs, but not new clusters, are chosen at periodic intervals to provide fairness, avoid single node failure, and rotate the energy draining role among sensor nodes within the cluster. For now, we assume that the total energy within a zone is the same. However, this assumption will be relaxed later.

Our scheme, GRASS, consolidates data aggregation and in-network processing at different levels of the virtual grid[2] (see Fig. 1. To be more specific, it is assumed that the set of the selected CHs in VGA perform two functions. First, they perform the first level of aggregation in each of their respective zones, hence they will be referred to as *Local Aggregators* (LAs). We assume that the data generated by all sensors in a zone is aggregated by the LA of this zone, and the rate of this data is known. Second, they relay information towards the BS. To further minimize energy consumption, we propose to use further levels of data aggregation using a subset of LAs. To be specific, we will first introduce algorithms that create a second-level of data aggregation through a subset of LAs called *Master Aggregators* (MAs), where an MA node will also act as an LA for

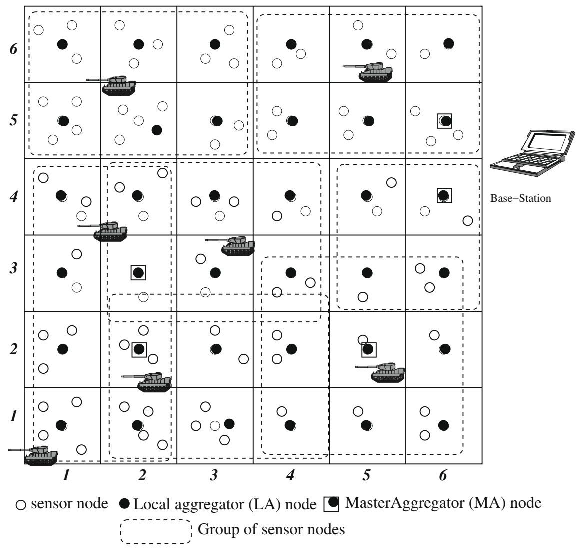


Fig. 1. The VGA clustering approach used in GRASS. Some nodes selected to act as aggregators at different levels.

the sensors in its own zone. We then extend these algorithms to generate a hierarchy (multi-level) of data aggregation points that further consolidate data before being forwarded to the BS. To be more concrete, sensors are at aggregation level 0, LAs are at aggregation level 1, MAs are at aggregation level 2, and so on. We study the tradeoffs for each case.

For a realistic scenario, we assume that a set of LAs which monitor the same phenomenon form a *group*, and the set of LAs is divided into, possibly overlapping, χ groups. That is, an LA may belong to more than one group. These groups may be known a priori in manual sensor deployment or can be constructed using message exchanges in random sensor deployment. Members of each group, S_g, $1 \leqslant g \leqslant \chi$, are sensing the same phenomenon, and hence their readings are assumed correlated. Thus, an LA node i that belongs to groups $j_1, j_2, \ldots, j_k$, will generate $m_i^{j_1}, m_i^{j_2}, \ldots, m_i^{j_k}$ data units every sensing period.[3] Such data units are to be delivered to the base station, BS. However, data units generated by different sensors in the same group may be aggregated by master aggregators, MA, on the route to BS. One MA may act as an aggregator for more than one group at the same time, but aggregating data belonging to different groups separately. For example, the MA in zone[4] (2,2) in Fig. 1 can act as an MA for the LAs in the group of zones {(1,1),(2,1),(1,2),(2,2),(1,3),(2,3),(1,4),(2,4)}, and the LAs in the group of zones {(2,1),(3,1),(4,1),(2,2),(3,2),(4,2)}.

It is assumed that packets generated by sensor nodes during a sensing interval are of variable lengths, and hence require different levels of energy for transmission and

[1] If the MAC layer is collision based, e.g., CSMA/CA, then it is assumed that the amount of data will be inflated in order to account for collisions and retransmissions. However, estimating the actual amount of data is beyond the scope of this paper since it does not deal with the MAC layer.

[2] Although all algorithms presented in this paper make use of the rectilinear virtual topology as their underlying transmission architecture, they can be easily extended to deal with any bounded-degree virtual topology.

[3] In the ILP and heuristic formulation, it will be assumed that these data units have constant lengths. In the simulation study, the data units will be taken from an exponential distribution, such that the mean packet length is equal to that used in the ILP or heuristics.

[4] A zone is referred to by the x–y coordinates shown in Fig. 1.

reception. We adopt the first order radio model in [5] which will be elaborated on in Section 6. However, for now we assume that the energy required for transmitting and receiving one byte are w_T and w_R, respectively. LA number i has a limited energy budget, E_i, which is used for sensing, computation and communication. It is assumed that the energy used for sensing and computation is negligible compared to the energy used for communication.

We consider two versions of this problem, and we formally define them as follows:

(a) **Joint Routing/MA Selection Problem in Two-level Scheme** (**RSP1**): Given a set of LAs, divided into, possibly overlapping, χ groups, the amount of traffic generated during a sensing period by each LA in each group, an external BS, and a maximum number of Master Aggregators (MAs) equal to M, find a subset of LAs of cardinality p ($p \leqslant M$) that will act as MAs for the LAs, and the routing from the LAs to the MAs, and from the MAs to the BS, such that the network lifetime is maximized. No MA can aggregate data already aggregated by another MA.

(b) **Joint Routing/Multi-aggregation Problem in Multi-level Scheme** (**RSP2**): Given a set of LAs divided into, possibly overlapping, χ groups, the amount of traffic generated during a sensing period by each LA in each group, and an external BS, find a suitable routing structure and aggregation points that gather data and report it to the BS such that the network lifetime is maximized. The maximum number of MAs should not exceed M, and an MA can aggregate data already aggregated by another LA or MA.

The network life time is defined formally as follows:

Definition 1. The *Network Lifetime* is defined as the time until the *first* CH node runs out of energy, i.e., the time until the first initially-populated zone becomes empty, or the zone has no sensor node with positive residual energy.

Note that our definition of lifetime is dependent on the total amount of energy in each zone. Recall that nodes in each zone take turns in serving as CHs according to our VGA clustering scheme. As the CH election is mainly based on sensor's remaining power, it is guaranteed that if a CH fails in a certain zone then that CH is the one with the highest power, and hence all other nodes in the zone have already failed. The definition of the network lifetime will be revisited in Section 3.3 for more practical scenarios.

It is to be noted that the above objective can be achieved by maximizing the minimum $\frac{E_i}{P_i}$ over all zones, i, where P_i is the power consumption of LA node $i \in N$ where N is the set of LAs in the network graph G, and E_i is the total amount of energy in zone i. However, although E_i is an input parameter, P_i is a variable which makes the problem non-linear. The problem can be transformed into a linear one by minimizing the maximum $\frac{P_i}{E_i}$ over all zones, i. If we assume a system in which the energy of a zone which is depleted is replenished with the same initial energy, i.e., a renewal process, then this last expression is the number of failures per second, or the rate of failure of zone i.

In the following sections, we present the algorithms of GRASS in details.

3. Exact algorithms: an ILP formulation

In this section, we present exact (optimal) algorithms for problems RSP1 and RSP2. For both problems, we introduce an Integer Linear Program (ILP) formulation to find optimal routes with data aggregation performed en route to minimize the maximum power consumption at each LA node, and hence maximize the network lifetime.

3.1. RSP1: two level routing/aggregation problem

In this section, we focus on problem RSP1, and study the selection of a set of LAs to serve as MAs such that the network lifetime is maximized. The problem of optimal selection of MAs is NP complete (Appendix A provides a complete proof of the NP-completeness of the problem RSP1). We now develop an Integer Linear Program (ILP) that finds the optimal selection of MAs satisfying the aforementioned objective. Let us label the Base-Station as node 0 and label the LA nodes as nodes 1 to n, where n is the total number of LAs (The variables and notations used in the ILP are defined in Table 1.)

Sometimes, only the summarized information is needed to serve the purpose of monitoring environmental events. In this case, the summarization (aggregation) function can take different forms such as duplicate suppression, averages, sums, minima, maxima, percentiles, etc. Our ILP can handle these different summarization functions. Indeed, the ILP can handle any linear function of the data to be aggregated. The ILP can also handle many non-linear functions which can be mapped to linear functions. As an example, we select the aggregation function to be the *maximum*, i.e., each MA selects the data unit that has the maximum size (length) coming from its constituent LAs source nodes. The maximum function is selected to represent cases when packets of variable lengths are collected by sensors, and hence variable energy is consumed in such type of communication. In one sense, selecting the maximum function represents a worst case scenario of collecting data with the highest communication cost.

We have noticed that it is sometimes possible to achieve the same level of energy consumption with different numbers of MAs. For example, we can aggregate four streams using either two MAs, or just one MA, which results in a smaller aggregated data, while using more hops to reach the MA as compared to the two MA case. In a case like this we opt to use the fewer number of MAs in order to reduce the computational requirements imposed on MAs due to aggregation. Also, in order to speed up the execution of the ILP solver, we limited the search space of the number of MAs to be about one half of the total number of LAs, which is the maximum number of second level aggregators under the two level aggregation assumption.

Objective function:

$$\text{Minimize}: \ \alpha * \rho + \beta * p.$$

The objective function minimizes the maximum zone failure rate over all zones while trying to also minimize the

Table 1
The ILP parameters and variables for problem RSP1.

	Input variables
N	The set of LAs in the graph G; with cardinality $n=\|N\|$ indexed by $1 \leqslant i \leqslant n$
M	The maximum number of MAs that can be allocated
F_{ij}	A binary indicator which is 1 if and only if nodes i and j are connected by a link
$P_{\max}$	The maximum power available for each LA node
α, β	Two weighting numbers (scalars) and $\alpha \gg \beta$
χ	Total number of LA groups in G
S_g	The set of LA nodes of group g, $1 \leqslant g \leqslant \chi$
m_i^g	The size (number of traffic units) of the packet sent by LA node $i \in S_g$ during one time period
E_i	Total amount of energy in zone i
Q	A very large number such that $(Q > \max_{g,1 \leqslant i \leqslant n}(m_i^g))$
w_T	A constant which represents the amount of energy spent in transmitting one byte (joules/byte)
w_R	A constant which represents the amount of energy spent in receiving one byte (joules/byte)
ρ	The maximum rate of failure among the set of all LA nodes
	Variables determined by the ILP
I_{ij}^g	A binary variable which is 1 iff LA node i of group g uses LA node j as its MA
p	Number of MAs allocated by the ILP ($p \leqslant M$)
Z_j^g	the maximum of all the lengths of the data units (packets) received by the jth MA node from members of group g, where $0 \leqslant Z_j^g \leqslant \max_{i, I_{ij}^g=1}(m_i^g)$
X_j^g	An auxiliary variable such that $X_j^g = I_{jj}^g * Z_j^g$, hence $0 \leqslant X_j^g \leqslant \max_{i,g}(m_i^g)$
R_{ij}^a	A binary indicator which is 1 if and only if LA node a is on the route from node i to node j
Y_{lk}^{ij}	A binary indicator; is 1 if and only if the traffic stream sourced at i and destined to j uses the link between node l and node k
P_i	Total power consumption by LA in zone i, which includes power consumption for possibility acting as an MA

number of MAs. The carefully selected values of α and β will ensure that the number of MAs will be reduced only if this does not result in an increase in the power consumption at individual nodes. Note that the network lifetime is given by $1/\rho$.

Subject to:

$$\sum_{j=1}^{n} I_{ij}^g = 1, \quad \forall g, i \in S_g, \tag{1}$$

$$\sum_{g=1}^{\chi} \sum_{j \in S_g} I_{jj}^g = p, \tag{2}$$

$$p \leqslant M, \tag{3}$$

$$I_{ij}^g - I_{jj}^g \leqslant 0, \quad \forall 1 \leqslant j \leqslant n,\ g, i \in S_g, \tag{4}$$

$$Z_j^g \geqslant m_i^g I_{ij}^g, \quad \forall 1 \leqslant j \leqslant n, 1 \leqslant g \leqslant \chi,\ i \in S_g, \tag{5}$$

$$X_j^g \geqslant Q I_{jj}^g - Q + Z_j^g, \quad 1 \leqslant g \leqslant \chi,\ 1 \leqslant j \leqslant n, \tag{6}$$

$$Xg \leqslant Z_j^g, 1 \leqslant g \leqslant \chi, 1 \leqslant j \leqslant n \tag{7}$$

$$\rho \geqslant P_i/E_i, \quad \forall i, \tag{8}$$

$$\rho \leqslant 1/T_{\min}. \tag{9}$$

Constraint (1) ensures that each LA node of group g is associated with only one MA node. Constraint (2) ensures that total number of allocated MAs for all groups is p. Constraint (3) guarantees that the number of MAs cannot exceed M. Constraint (4) ensures that an LA node i of group g will use LA node j as its MA only if LA j has been selected to act as an MA j. Constraint (5) ensures that the maximum-sized packets are sent to the BS taken over all the packets generated by the LAs of the groups served by the jth MA. Constraints (6) and (7) (together with the minimization of the objective function) set X_j^g to the amount of traffic routed from the jth MA to the BS, i.e., the product $I_{jj}^g * Z_{jj}^g$. Constraint (8) finds the maximum failure rate over all zones (it will be shown below how to evaluate the power consumption at each LA in order to calculate this rate). Constraint (9) guarantees a minimum zone lifetime and limits the maximum power consumption of any node.

The power consumption at each node is also dependent on the determination of the actual routing of the data traffic, i.e., we must find the power consumed by each LA source node when participating in routing data over G. The following additional set of constraints are required for performing route computations needed to find the values of R_{ij}^a. The following two constraints ensure that for the connection going from node i to node j, no traffic is coming in (going out) the source i (destination j), respectively

$$\sum_{k, F_{ki}=1, k \neq i} Y_{ki}^{ij} = 0 \quad \forall i, j; \qquad \sum_{k, F_{jk}=1, k \neq j} Y_{jk}^{ij} = 0 \quad \forall i, j.$$

The following two constraints ensure that the connection traffic between i and j is originating (terminating) at i(j), respectively

$$\sum_{k, if F_{ik}=1, k \neq i} Y_{ik}^{ij} = 1 \quad \forall i, j; \qquad \sum_{k, if F_{kj}=1, k \neq j} Y_{kj}^{ij} = 1 \quad \forall i, j.$$

The following constraint preserves the continuity of connection traffic on one of multiple possible routes

$$\sum_{k, if F_{kx}=1, k \neq x, k \neq j} Y_{kx}^{i,j} = \sum_{j, if F_{xk}=1, j \neq x, j \neq s} Y_{xk}^{i,j} \quad \forall x, (x \neq i, j), i, j. \tag{10}$$

Note that although the above can result in loops, the minimization of power consumption will prevent the formation of loops, since any loop will lead to greater power consumption.

The following two constraints determine the value of R_{ij}^a, i.e., if LA a is on the route(s) between LAs i and j.

$$R_{ij}^a \geqslant \sum_k Y_{ka}^{i,j}/Q \quad \forall i, j, a, i \neq j, i \neq a, j \neq a, \tag{11}$$

$$R_{ij}^a \leqslant \sum_k Y_{ka}^{ij} \quad \forall i, j, a, i \neq j, i \neq a, j \neq a. \tag{12}$$

The power consumed by node a is given by

$$P_a = \sum_{j,g=1} w_T I^g_{aj} m^g_a + \sum_{i,j,g} (w_T + w_R) I^g_{ij} R^a_{ij} m^g_i + \sum_{g=1}^{\chi} \sum_{i \in S_g} w_R I^g_{ia} m^g_i + \sum_{g=1}^{\chi} w_T X^g_a + \sum_{j,g,j \neq a} (w_T + w_R) X^g_j R^a_{j0}. \quad (13)$$

The five terms in the above equation refer to: (1) the power consumed by data sourced out from LA node a; (2) the power consumed by node a while relaying for other LAs towards an MA node, including power consumed in reception and transmission of data units; (3) the power consumed by an MA in receiving unaggregated data units from the LAs; (4) the power consumed by MA a in transmitting the data units after aggregation; and (5) the power consumed while relaying for other LAs towards the BS, which includes power for reception and transmission.

Note that the conjunction $I^g_{ij} \wedge R^a_{ij}$ in Eq. (13) is mapped to a linear form by defining $U^{ga}_{ij} = I^g_{ij} \wedge R^a_{ij}$ and using the following two constraints, which is a standard mathematical programming modeling technique:

$$U^{ga}_{ij} \geqslant I^g_{ij} + R^a_{ij} - 1, \quad (14)$$
$$U^{ga}_{ij} \leqslant (I^g_{ij} + R^a_{ij})/2. \quad (15)$$

Finally, the term $X^g_j R^a_{j0}$ is a non-linear term. We introduce the following approach to linearize this product. We define $V^{ga}_{jj} = X^g_j R^a_{j0}$, and use the following two linear equations to evaluate V^{ga}_{jj}:

$$V^{ga}_{jj} \geqslant QR^a_{j0} - Q + X^g_j, \quad (16)$$
$$V^{ga}_{jj} \leqslant X^g_j. \quad (17)$$

This is similar to the approach used to evaluate X^g_j in Eqs. (6) and (7). Using Eqs. (14)–(17), the power consumed by LA node a can now be expressed as:

$$P_a = \sum_{j,g=1} w_T I^g_{aj} m^g_a + (w_T + w_R) \sum_{i,j,g} U^{ga}_{ij} m^g_i \sum_{g=1}^{\chi} \times \sum_{i \in S_g} w_R I^g_{ia} m^g_i + \sum_{g=1}^{\chi} w_T X^g_a + \sum_{j,g,j \neq a} (w_T + w_R) V^{ga}_{jj}. \quad (18)$$

It should be noted that the number of variables in the above formulation is $O(n^3\chi + n^2E)$, where E is the number of edges in the network graph. This is based on the number of the auxiliary variables U^{ga}_{ij}, and the number of binary indicators, Y^{ij}_{lk}, respectively.

3.2. RSP2: multi-level routing/aggregation problem

In this section, we extend the ILP presented in the previous section to handle multiple levels of data aggregation. The problem of optimal selection of multiple levels of aggregation is harder than the two level aggregation scheme. In other words, The RSP2 problem is harder than RSP1, and is therefore also NP-complete (the verification of a certificate for RSP2 can also be performed in polynomial time). However, we anticipate that multilevel aggregation will result in further reduction of energy dissipation, and hence prolong the lifetime more than the two-level data aggregation scheme. Our objective here is similar to our objective in Section 3.1, which is to find optimal routes and a set of aggregation points for each group such that the network lifetime is maximized. Since with multiple levels of aggregations the maximum number of MAs is $n-1$, where n is the total number of LAs, we did not impose any upper bound on the number of MAs, as this will not provide any significant reduction in the search space. We have also decided not to minimize the number MAs in order to explore the full limit of aggregation, and since the number of MAs is already much larger than that under two level aggregation.

The ILP formulation presented in this section is generic and can handle arbitrary data aggregation factors and arbitrary traffic streams. We define r^g_k as the fraction of original traffic of a source left after aggregation, when data streams from k members of group g are aggregated. Hence the data aggregation (or reduction) factor is $(1 - r^g_k)$. The variables and notations used in the ILP of RSP2 are defined in Table 2, while other related notations are already defined in Table 1. Below, we present the ILP:

Objective function:

Minimize ρ.

The objective function minimizes the maximum zone failure rate

Subject to:

The following two constraints find the maximum power consumption over all LA nodes, while respecting the maximum power consumption limit of any LA node,

$$\rho \geqslant P_i/E_i, \quad \forall i, \quad (19)$$
$$\rho \leqslant 1/T_{\min}. \quad (20)$$

Table 2
The ILP parameters and variables for problem RSP2.

	Input variables
r^g_k	Fraction of original traffic left after aggregation, when data streams from k members of group g are aggregated such that $r^g_k \geqslant r^g_{k+1}$ and $r^g_k * k \leqslant r^g_{k+1} * k + 1$, $0 \leqslant k \leqslant \lvert S_g \rvert$ with $r^g_1 = 1$ and $r^g_0 = 0$
Q	A very large number such that $(Q > \lvert S_g \rvert \max_{g,1 \leqslant i \leqslant n}(m^g_i))$
$e_{i,j}$	The link weight between LA nodes i and j, which is assumed *symmetric* in graph G
	Variables determined by the ILP
$X^g_{i,j}$	Number of data units coming from members of group g and sent by node i to node j, and after aggregation if any
P_i	The amount of power utilized at LA node i
$T^{s,g}_{ij}$	A binary indicator; is 1 if and only if the traffic stream sourced at node s of group g uses the link between nodes i and node j
$I^{g,k}_{ij}$	A binary indicator; is 1 if and only if the number of LA source nodes in group g that are sending data through the link from node i to node j is $\leqslant k$
M^g_{ij}	A binary indicator; is 1 if and only if the link going from node i to node j carries the traffic by at least one source of group g

The following constraint ensures that the power consumed at each LA node is sufficient to send the amount of required traffic to the other nodes.

$$P_i \geqslant \sum_g \left(\sum_{j,\ \text{if}\ F_{ij}=1} X_{ij}^g w_T + \sum_{j,\ \text{if}\ F_{ji}=1} X_{ji}^g w_R \right) \quad \forall g, i. \tag{21}$$

The following two constraints ensure that if K is the number of members of group g sending data from node i to node j, then the value of $I_{ij}^{g,k}$ is 1 for all the values of k that are greater than or equal to K; otherwise it will be 0

$$\sum_{k=0}^{|S_g|} I_{ij}^{g,k} + \sum_{s\in S_g} T_{ij}^{s,g} = |S_g| + 1, \quad \forall g, j, i, \quad \text{if } F_{ij} = 1, \tag{22}$$

$$I_{ij}^{g,k} \leqslant I_{ij}^{g,k+1}, \quad \forall g, 0 \leqslant k \leqslant |S_g| - 1, i, j, \quad \text{if } F_{ij} = 1. \tag{23}$$

The following two constraints together give the exact amount of traffic sent by node i to node j, once node i has aggregated the data coming from members of group g:

$$X_{ij}^g \geqslant I_{ij}^{g,k} * r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g}, \quad \forall g, j, k, i, \quad \text{if } F_{ij} = 1, \tag{24}$$

$$X_{ij}^g \leqslant r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g} + Q * I_{ij}^{g,k} \left(k - \sum_{s\in S_g} T_{ij}^{s,g} \right), \quad \forall g, j, k, i, \quad \text{if } F_{ij} = 1. \tag{25}$$

However, constraints (24) and (25) are non-linear. We introduce an approach to linearize these constraints, which is illustrated by the following two equations:

$$X_{ij}^g \geqslant Q * I_{ij}^{g,k} - Q + r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g}, \quad \forall g, j, k, i, \quad \text{if } F_{ij} = 1, \tag{26}$$

$$X_{ij}^g \leqslant r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g} + Q|S_g| * (1 - I_{ij}^{g,k}) + Q * k * I_{ij}^{g,k} - Q * \sum_{s\in S_g} T_{ij}^{s,g}, \quad \forall g, j, k, i, \quad \text{if } F_{ij} = 1. \tag{27}$$

Only when $k = K$ that the above equations will reduce to:

$$r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g} \leqslant X_{ij}^g \leqslant r_k^g * \sum_{s\in S_g} m_s^g T_{ij}^{s,g},$$

which is the desired aggregated bandwidth. This holds since $I_{ij}^{g,k} = 1$ and $\left(k = \sum_{s\in S_g} T_{ij}^{s,g}\right)$ for $k = K$.

The following three constraints ensure that the aggregated traffic streams will not be split on the way to the BS:

$$M_{ij}^g \geqslant \sum_{s\in S_g} T_{ij}^{s,g}/Q \quad \forall g, i, j, \quad \text{if } F_{ij} = 1, \tag{28}$$

$$M_{ij}^g \leqslant \sum_{s\in S_g} T_{ij}^{s,g} \quad \forall g, i, j, \quad \text{if } F_{ij} = 1, \tag{29}$$

$$\sum_{j,\ \text{if}\ F_{ij}=1} M_{ij}^g \leqslant 1, \quad \forall g, i. \tag{30}$$

The guarantee of a minimum lifetime of an LA node is highly dependent on the determination of the actual routing of the data traffic, i.e., we must find the power consumed by each LA source node when participating in routing data over G. The following additional set of constraints are required for performing route computations.

The following two constraints ensure that for the traffic from source s to the base station, 0, no traffic is going in (going out) the source s (destination 0), respectively

$$\sum_{i,\ \text{if}\ F_{is}=1, s\neq i} T_{is}^{s,g} = 0; \quad \sum_{j,\ \text{if}\ F_{0j}=1, j\neq 0} T_{0j}^{s,g} = 0 \quad \forall g, s \in S_g.$$

The following two constraints ensure that the traffic from s and 0 is originating (terminating) at s (0), respectively

$$\sum_{j,\ \text{if}\ F_{sj}=1, s\neq j} T_{sj}^{s,g} = 1 \quad \sum_{i,\ \text{if}\ F_{i0}=1, s\neq 0} T_{i0}^{s,g} = 1 \quad \forall g, s \in S_g.$$

The following constraint preserves the continuity of connection traffic on one of multiple possible routes

$$\sum_{i,\ \text{if}\ F_{ix}=1, i\neq x, s\neq i} T_{ix}^{s,g} = \sum_{j,\ \text{if}\ F_{xj}=1, j\neq x, j\neq s} T_{xj}^{s,g} \quad \forall x, g, s \in S_g, \quad (1 \leqslant x \leqslant n,\ x \neq s). \tag{31}$$

Finally, we point out that the number of variables used in this formulation is $O(n^2\chi + \chi E \max_g |S_g|)$, which is based on the number of X_{ij}^g and $T_{ij}^{s,g}$ (or $I_{ij}^{g,k}$), respectively.

3.3. Extension to the network partitioning condition

In the above formulation, it was assumed that the network lifetime is limited by the failure of any single zone, i.e., all sensors in a zone. In some situations, this may not be a practical condition. A zone may fail, but the rest of the network may still operate. However, we realize that the network lifetime may be limited by its capability to deliver data to the base station. Therefore, in this section we introduce an alternative definition of the network lifetime:

Definition 2. The *Network Lifetime* is defined as the time period from the instant when the network starts functioning to the instant when the network becomes partitioned, i.e., when the *first CH node that acts as a router for other CHs* runs out of energy.

This requires the identification of those zones in which the LA serves as a router, which is done through the following binary variable:

:b_i: a binary variable that is equal to 1, if and only if the LA in zone i acts as a router.

b_i can be set to 1 if the energy consumed by the LA in zone i exceeds the energy required for the transmission of data generated within this zone.

This indicates that this zone is involved in routing other data. Therefore, the following two constraints will evaluate b_i:

$$b_i \leqslant \left(P_i - \sum_{g=1}^{\chi} w_T * m_i^g \right) \cdot A \quad \forall i, \tag{32}$$

$$b_i \geqslant \frac{P_i - \sum_{g=1}^{\chi} w_T * m_i^g}{A} \quad \forall i, \tag{33}$$

A in these two equations is a very large number which is chosen such that whenever $P_i > \sum_{g=1}^{\chi} w * m_i^g$, the right-hand side in Eq. (32) is greater than or equal to 1, and

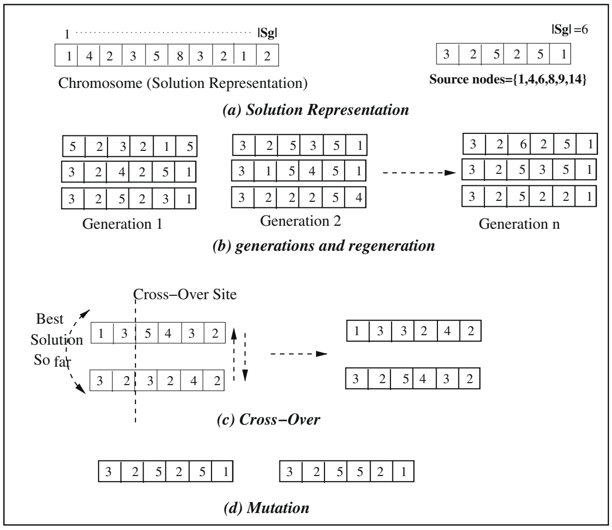


Fig. 2. Representation of solution in multi-level aggregation scheme in the form of routing structure.

the right hand side in Eq. (33) is a non-negative number less than or equal to 1. Finally, Eq. (19) is replaced by the following constraint

$$\rho \geqslant P_i/E_i - (1 - b_i) \cdot A, \tag{34}$$

where A in this case is chosen to be greater than the largest value of P_i. This constraint means that ρ is an upper limit on the rate failure only for those zones which also provide data relay service to other zones.

3.4. A note on optimality

It is obvious that the optimal approaches above set up static routes which are used for the entire network lifetime. Therefore, an argument may be raised that repeating the ILP periodically, or according to different criteria, may result in extending the network lifetime. However, we note that the objectives of the optimal formulations to problems RSP1 and RSP2 above is to minimize the rate of energy depletion per zone. The ILP, in fact, attempts to equate those rates, and it turns out that in the numerical results the energy depletion rates are almost equal, and if there are differences, the differences are very small. Such an observation has also been made in [28]. Therefore, it is very difficult to achieve any better performance using any different approach. Moreover, repeating the ILP is not practical because of its high time complexity.

4. Heuristic approaches

In the previous section, we presented optimal solutions for both **RSP1** and **RSP2** problems using an ILP formulation. In practice, WSNs are large networks with hundreds of nodes, which the ILPs cannot handle with reasonable computational facilities. Therefore, a more efficient, albeit less optimal, scheme is highly desirable. In this section, we present two heuristic approaches to find near optimal solutions for both **RSP1** and **RSP2** problems. The first heuristic approach is based on genetics algorithms [24]. The other heuristic approach aims at balancing power consumption among sensor nodes in order to prolong the network lifetime in both the two- and multi-level approaches. We have also designed and tested other heuristics, e.g., a modified k-means clustering algorithm and a simple greedy algorithm for the two-level aggregation scheme (see [22]). However, we will neither present the details nor the results for k-means and the greedy approach since they are inferior to the heuristics presented in this section especially for the multi-level aggregation scheme presented in this paper.

4.1. A genetic algorithms approach

We developed a genetic algorithm strategy to solve both the RSP1 and RSP2 problems. The major step in GA is to find an efficient way to represent the solution. The detailed operation of GA is well-known and can be found in [24]. For the extended problem (RSP2), our solution representation of an individual is shown in Fig. 2. For each group g, where $1 \leqslant g \leqslant \chi$, the solution is represented as a string of length $|S_g|$, where $|S_g|$ is the number of LAs in group g. The ith cell in the string contains the route number that will be used by the ith source of group g, which has an integer value between 0 and $R^i_{\max}$, $R^i_{\max}$ denotes the number of routes from the ith source to BS. We assume that this number is known and can be found by any route discovery technique, e.g. reactive protocols. Hence, the *individual* is a routing structure for each group. First, we generate an initial population of randomly created, and thus different, routing structure for each group g. The generated population is subjected to the typical GA evolution process. Fig. 2b–d shows the process of reproducing generations, crossover, and mutation, respectively.

Due to the lack of space, we omit the details of the genetic algorithms approach, which are available in [38]. Results based on this algorithm will be shown in Section 6.

4.2. The balanced power consumption heuristic approach

In this section, we present a heuristic that balances the power consumption at sensor nodes for the sake of maximizing the network lifetime in a multi-level data aggregation schemes.[5]

The heuristic, called Clustering-Based Aggregation Heuristic (CBAH), is inspired by some observations taken from the ILPs developed earlier. After running the ILPs for many scenarios, we noticed that, when the amount of energy in the non-empty zones is the same, the power consumed by different LA nodes is almost always the same. In other words, if we define

$$\Delta = \max_{i,j}(|P_i - P_j|), \quad \forall i,j \in N, i \neq j$$

as the maximum absolute difference in the power consumption levels between any two LA nodes, i and j, in the graph G, then the value of Δ should almost always equal to zero. This indicates that the optimal solution also achieves balanced power consumption among all LA nodes in our scheme. And, in the case of unequal amounts of energy for different zones, the optimal solution equalizes the zone failure rates. Therefore, we make use of this fundamental observation to develop a simple and efficient heuristic, namely, CBAH, which can be used to solve for large WSNs.

The heuristic CBAH will find routes on the graph G for each source node of each group such that power consumption in the network is balanced. In order to achieve balanced power consumption and extend network lifetime, a source node should select a route that minimizes the maximum total power consumption (P_i) at each individual node i in the route taken over all feasible routes for that source node, while allowing for data aggregation en route. To achieve this goal, CBAH will use the information about the total power consumption experienced at each LA node so far in addition to the required transmission power.[6] Note that a node with high total power consumption or low remaining energy should be avoided when performing routing.

To be formal, let each source node s in each group g, $1 \leqslant g \leqslant \chi$ search for a route over G. Indeed, many disjoint and non-disjoint routes to the destination (BS) can be found in G for any source s. However, to limit the complexity of the problem, we consider routing through a limited number of *semi-disjoint* paths, where those paths share the least common nodes. The term semi-disjoint will be further defined below. Let $\mathscr{R}$ be the set of these semi-disjoint paths. Denote by $N(r_s)$ the set of nodes on route r_s, $r_s \in \mathscr{R}$ for source node s. CBAH will select for each source node s of each group g, $1 \leqslant g \leqslant \chi$, a route, r, among all possible $\mathscr{R}$ routes such that,

$$r = \arg\min_{r_s \in \mathscr{R}} \{\max_{i \in N(r)}(P_i)\}.$$

The objective of selecting such a route is to smooth the use of the battery of each node and extend its lifetime. If LAs are heterogeneous, and start with different power budgets, then the above selection criterion can be revised to

$$r = \arg\max_{r_s \in \mathscr{R}} \{\min_{i \in N(r)}(B_i)\},$$

where B_i is the remaining power budget at LA node i. This will achieve the same objective of maximizing the network lifetime.

Now, we describe the operation of CBAH. A high level description of CBAH is shown in Fig. 3. Let A be the adjacency matrix of the virtual graph G, where the entry (i,j) is 1 if there is a link from LA node i to its adjacent LA node j; otherwise it is 0. Let $P_i = 0$ be the initial total power consumption within each LA node i in graph G. Also, define *aggregator*$[g][n]$ as the set of aggregator nodes which is initially empty. Let *routes*$[g][s]$ be the set of semi-disjoint routes from source s to the BS, which is initially empty. The heuristic CBAH will cycle through each group and for each source in a group, it finds a route to BS (node 0) such that power consumption at the network is balanced, and hence the network lifetime is prolonged. The set of discovered routes for each source node is stored in a list called *routes*$[g][s]$. The list *routes*$[g][s]$ is three-dimensional list that stores groups, sources of each group, and the discovered routes for each source node. The details are as follows.

A source s starts finding its available routes to BS by invoking a function called FIND-ROUTE(s). Fig. 4 shows the operation of FIND-ROUTE(s). The first step is to inspect the neighboring LA nodes of s. The set of neighbors are stored in a list called the *candidate-list*, which will be ex-

```
Input: A, n, m_i^g, χ, S_g;
Output: A set of routes and aggregation points to maximize the network lifetime.;
Algorithm CBAH:
Initialize;
Label all LA nodes as "un-visited";
Set P_i=0.0; 1 ≤ i ≤ n, aggregator[g][n]={φ}, N_r^s={φ},
routes[g][s]={φ}, R_max=4, route-found=false, num-sources=0;
Define set of struct candidate-list[], new-candidate-list[], nbr,
element, path-element;
for (g=1 to χ) do
    for (j=1 to |S_g|) do
        r=1 /*for each source node j.*/;
        s=get-ID(j);
        while (r ≤ R) do
            route-found = FIND-ROUTE(s);
            if (route-found) then
                rt=backtrack(s,g);
                Insert-route(rt,g,s);
            end
            r=r+1;
        end
                    (j, routes[g][i], num-sources);
        Label all nodes in G as "un-visited";
        num-sources++;
    end
    num-sources=0;
end
```

Fig. 3. A high level description of the CBAH heuristic for RSP2 problem.

[5] We developed a two level aggregation approach for the RSP1 problem, which we call the Load Balancing Approach (LBA), which is regarded as a special case of the heuristic of this section.

[6] This also applies to the case when sensor nodes have different initial energy levels (heterogeneous network). In this case, the remaining energy at each node can be used.

Input: s;
Output: if a route toward the BS is found, the 'route-found' variable will be true.;
Function FIND-ROUTE(s);
candidate-list[]=$\{\phi\}$;
new-candidate-list[]=$\{s\}$;
while *(0 $\not\subseteq$ new-candidate-list)* **do**
  /*BS is not reached*/;
  candidate-list $\leftarrow$ new-candidate-list;
  new-candidate-list=$\{\phi\}$;
  for *(element $\in$ candidate-list)* **do**
    for *(each nbr of element)* **do**
      /* nbr is obtained from A*/;
      /**The nbr node is located above, below, at the left or at the right of a node.**/;
      if *(nbr is "un-visited") || (element is "un-visited" && nbr is "visited")* **then**
        set this neighbor as "visited";
        new-candidates-list=new-candidates-list $\cup$ $\{nbr\}$;
        set predecessor of nbr to element;
      end
      if *(nbr == 0)* **then**
        return $true$;
      end
    end
  end
end

Fig. 4. A high level description of the function FIND-ROUTE in CBAH.

panded during the route search process. Each member of this list is used to initiate the next hop during the route search process. The importance of the *candidate-list* is that it allows one to quickly select the next LA node needed to expand the route search for a certain source node. In each expansion step, the contents of *candidate-list* are refreshed by using an auxiliary list called *new-candidate-list*, which contains the nodes that will be used for the next step in route search. Each time, one candidate is pulled from the list and its four (potential) neighbors are examined for possible expansion of the route search. Whenever a node is inserted in the *new-candidate-list*, CBAH keeps a pointer to the predecessor node, which is required to backtrack the route(s) to the source node in case this node falls on the selected route. By repeating the route discovery phase, CBAH may find multiple disjoint paths by marking nodes which have been *visited*; thus forbidding the visited nodes from being part of more than one route for the same source node. This approach may restrict the number of routes found for each source node, especially if the graph of LAs is sparse. An extension to this approach is to allow the already visited nodes to be inspected again, i.e., an already visited node can still be visited from nodes that are only labeled as un-visited. This allows checking for more routes, and hence find better *semi-disjoint* routes, if available. In our implementation of CBAH, we used this latter option. In CBAH, *semi-disjoint* routes means those routes that share the least common nodes. As described in Fig. 5, when the destination node (node 0) is reached, the function *backtrack* (s,g) traverses each discovered route back to the source node by using the pointers set up earlier.

After finding the set of routes for each source s in a group g (routes[g][s]) and in order to determine what route among these routes the source s will use, CBAH invokes a function called select-route($\cdot$), and described in Fig. 5. During this process, CBAH interactively checks the allocated routes of other sources of the same group and enforces aggregation at common intermediate LA nodes, i.e., determine aggregator nodes for that group, using another function called *compare-routes*()) that is depicted in Fig. 6. The function *compare-routes*() starts by finding the best aggregation points, which sometimes can be the closest node to the source. The route will be trimmed to reach the aggregator point only since the rest of the path to BS is saved in the aggregation point. The set of these aggregation points will then be saved in the list (*aggregator*[g][s]). The aggregated data stream follows the same path till another source node of the same group joins this path at a later point, where the process repeats, i.e., further aggregation is performed at this aggregator node, and the new data aggregate is sent along one route to the BS. This means that once data of multiple streams is aggregated at a certain node, no splitting of the aggregate is allowed. That is, CBAH will fuse routes of the source nodes at a shared common node and then unify the rest of the route. This last process is executed in the function *compare-routes*().

What is left is the case when a source node falls in an overlapping region, i.e., belongs to more than one group. In this case, CBAH holds a registry for those source nodes in order to distinguish between their sent data. Once a route r is selected for a source node s, the function *Update-Power*(s,$N(r)$) which is called from within the function *select-route*() will update the amount of total power consumption at each node i in the route r, $P_i, i \in N(r)$ by adding

Input: s;
Output: A route from s to BS is selected.;
Function SELECT-ROUTE(s, routes[g][s], num-sources);
begin
select r_s= arg $\min_{r \in routes[g][s]}(\max_{k \in N_r^s}(P_k))$;
if *(num-sources > 1)* **then**
  aggregator[g][s]= **COMPARE-ROUTES**(r_s,routes[g][ ]);
end
Compute power consumption;
m_r=$m_s^g * w$ /*w is as defined in ILP*/;
N_r^s= get-route-nodes(s,r_s,$routes$[][]);
Update-Power(N_r^s,m_r) /**including relay power**/;
N_r^s=$\{\phi\}$;
end;

Fig. 5. A high level description of the function select-route in CBAH.

Input: r_s;
Output: Aggregator node for source s.;
Function COMPARE-ROUTES(r_s, routes[g][]);
begin
List aggregators = $\{\phi\}$;
/* *find common points with routes of other sources within the same group*/;
aggregators = Find-common-points(rs, routes[g][]);
if (aggregators-size() > 1) /**if more than one common point are found, then select the closest one**/;
myaggregator = select-closest(aggregators);
/**Trim the source route to reach the aggregator point only**/;
route-trim(r_s);
return myaggregator;
end;

Fig. 6. A high level description of the function compare-route in CBAH.

the link cost $e(i,j)$ to the current value of P_i. Note that each node, which acts as an aggregator, consumes additional power when relaying aggregated data for other nodes. CBAH updates those nodes as well by inspecting the list (*aggregator*[g][s]). After the routes and the set of aggregation points for each source in a certain group are selected, all LA nodes must be cleaned by resetting the label of each node to *un-visited* before another group is served. At the end of CBAH execution, a set of routes as well as a set of aggregator nodes are determined for each source node in each group of the graph G. The algorithm CBAH has a worst case time complexity of $O(n\chi R_{\max})$, and a space complexity of $O(n|S_g|)$.

5. Energy-delay tradeoff analysis

Although data aggregation results in fewer data transmissions, there is a tradeoff between energy savings and the delay due to the aggregation process. This potential delay may occur because data from closer sources may have to be buffered at an intermediate MA node in order to be aggregated with data coming from sources that are farther away. Therefore, the amount of delay introduced by aggregation needs to be evaluated, and an application dependent maximum delay should be enforced. Recall that we are assuming periodic sensing with the same period for all sensor nodes. For simplicity, we assume that all sensors in the same group are synchronized, and all measurements are taken by all sensors in the same group at the same time. However, asynchronous operation can be accounted for by adding a time factor that accounts for this mode of operation.

In our scheme, the aggregation delay occurs at two levels, local and global. In the local aggregation process, delay can be considered negligible since source nodes are in the same zone and they are able to communicate with their peer LA nodes directly. Hence, the aggregation delay is mainly due to the global data processing at farther aggregation points. To find the total delay, however, the aggregation delay must be added to the total processing and communication delays required to reach the BS from that MA node. Nevertheless, we are only interested in finding the aggregation delay, that is, the delay incurred by reporting data from different LA source nodes located at different distances from a certain aggregation node. Note that processing delays at aggregation points will be small when compared to the delay incurred in communicating data to the BS. We will now analyze the aggregation delays associated with the two-and multi-level aggregation schemes presented in the previous sections.

5.1. Aggregation delay under two-level aggregation

Assume that the ILP, or any of the approximation algorithms for the two level scheme have been executed and the set of MAs, $\mathscr{M}$, have been allocated in the two-level aggregation scheme and the set of routes and the set of aggregation points for each group have been determined. Assume that S_g^j is the set of LA source nodes in group g, $g = 1, 2, \ldots, \chi$, associated with the jth MA node, $j = 1, 2, \ldots, p$. Let d_g^{ij} be the distance (in terms of the number of hops) of the path going from the LA source node $i \in S_g^j$ to its assigned jth MA node in the graph G. Let $D_{\max}^j$ and $D_{\min}^j$ be the maximum and the minimum distance over all values of d_g^{ij} with respect to jth MA node, respectively, i.e.,

$$D_{g,\max}^j = \max_{i \in S_g^j} d_g^{ij} \quad D_{g,\min}^j = \min_{i \in S_g^j} d_g^{ij}.$$

- **Duplicate suppression**:
  If the aggregation function is duplicate suppression, the aggregation delay incurred by group g, T_g^{ds}, will be simply the maximum time needed to receive the first unique packet at an MA node calculated over the set of all MA nodes. Each node only passes the first unique packet and suppresses subsequent packets with identical sequence numbers. Hence,

  $$T_{ds} = \max_{j \in \mathscr{M}} [\min_{i \in S^j} d_{ij}].$$

- **Max, min, and average aggregation**:
  In the case of general aggregation functions (e.g. maximum, minimum, average), the aggregation delay incurred, T_{ga}, will be the maximum of the difference in number of hops between an LA source node and its assigned MA node evaluated over all pairs of LA source-MA nodes. Hence,

  $$T_{ga} = \max_{j \in \mathscr{M}, g \in \chi} [D_{g,\max}^j - D_{g,\min}^j].$$

5.2. Aggregation delay under multi-level aggregation

In the multi-level aggregation scheme, the latency will be proportional to the number of hops between the data aggregation point from the farthest LA source node reporting data to the last aggregation node in the aggregation tree for each group taken over the set of all groups. Let the aggregation function be the duplicate suppression. Let A_f^g be the final aggregation point for all traffic coming from group $g \in \chi$. Let (t^g) be the maximum time required to reach node A_f^g from all LA sources in group g, and define $t_{\max}$ as the maximum delay time due to aggregation taken over all groups in the graph G. Let $D_g(i)$ be the delay time taken by data coming from ith LA source node of group g to reach node A_f^g. The delay $D_g(i)$ is computed by finding the number of hops taken by LA source node i, $i \in S_g$ to reach the last aggregation point, A_f^g, for the group g. Then, the overall maximum aggregation delay is given by

$$t_{\max} = \max_{g \in \chi}(t^g) \quad \text{and} \quad t^g = \max_{i \in S_g}(D_g(i)).$$

In the case of general aggregation functions (e.g. maximum, minimum, average), the aggregation delay incurred is evaluated as in the 2-level scheme except that the minimum and maximum are found with respect to the last aggregation point.

6. Performance evaluation

The performance of the algorithms of GRASS were tested with various experimental scenarios which were simulated using the NS-2 simulator [40]. Each experiment corresponds to a random placement of sensors in a fixed network area. We assume a single Base-Station attempting to gather information from a number of data sources in the network area. The location of the Base-Station can be arbitrarily chosen. Unless stated otherwise, the BS is located at lower edge of the grid (0,0). We randomly place sensor nodes in a 50 m × 50 m square field while always insuring that the initial distribution of sensor nodes always results in a connected graph, as will be explained below. We also experimented with larger sensor fields to test the performance of various heuristics in large networks. It is assumed that the sensing range is the same as the transmission range which was set to a default value of 20 m. The sensor field is divided into the appropriate number of zones, which is 30. We consider four scenarios corresponding to the distribution of sensors in the sensing field, which result in z nonempty zones (clusters) forming a connected virtual graph G. In the four scenarios, n takes values of 6, 8, 10, and 15, respectively. In each nonempty zone, there are on average 10 sensor nodes monitoring the area of that zone. We assume that sensors generate data packets of variable sizes such that the packet size is exponentially distributed with mean value of 1000 bits. In another setting, we fixed the packet size generated by all nodes for the sake of comparison with other schemes. In all settings, except for the sake of comparison to other schemes, the aggregation function used by GRASS is based on taking the packet size with the maximum length.

Each sensor i has a battery with finite, non-replenishable energy, which was set to an initial energy of 2 Joules. Whenever a sensor transmits or receives a data packet, it consumes some energy from its battery. The base station has an unlimited amount of energy. The choice of the MAC protocol can completely dominate energy consumptions. We assume that energy-conscious protocols like PAMAS [25] or TDMA-based MAC [5] are used for long-lived sensor networks. Our energy model for the sensors is based on the first order radio models [5,26] in which a fixed amount of energy is spent in transmitting and receiving a packet in the electronics, and an additional amount proportional to the distance between tow nodes is spent in transmitting a packet. The radios can perform power control and hence use the minimum required energy to reach the intended recipients. Due to attenuation with distance, an energy loss model with d_{ij}^2 is used for relatively short distances, where d_{ij} is the distance between sensor nodes i and j. More precisely, a radio dissipates $E_{\mathrm{elec}} = 50\ \mathrm{nJ/bit}$ to run the transmitter or receiver circuitry and $\varepsilon_{\mathrm{amp}} = 100\ \mathrm{pJ/bit/m^2}$ for the transmitter amplifier. Thus, the energy consumed by a sensor node i in receiving a 1000-bit data packet is $1000 * 50\ \mathrm{nJ/bit} = 50\ \mu\mathrm{J}$, while the energy consumed in transmitting a data packet from sensor i to sensor j is given by $T = 50\ \mu J + 100\ \mathrm{nJ/m^2} \times d_{ij}^2$. A link transmission rate of 1 Mbps is assumed. We make the assumption that the radio channel is symmetric such that the energy required to transmit a message from LA node A to LA node B is the same as the energy required to transmit a message from LA node B to LA node A. As for delay on a link, it can be calculated as units of time. On a 1 Mbps link, a 1000 bit message can be transmitted in 1ms. We assume that each unit of delay corresponds to 1ms time. Hence, the delay is 1 unit for each 1000 bit message transmitted. The BS is placed at the middle of the network area.

Table 3
Lifetime extension ratio (L) for different 2L aggregation approaches, and for different values of n and M, $\chi = 3$.

n	M	No aggregation	GA	LBA	*ILP*
5	3	1	2.45	2.83	3.01
10	5	1	3.97	4.33	5.03
15	7	1	3.75	4.86	5.69
20	9	1	4.25	4.30	4.98

We ignore edge effects where smaller zones on the boundary of the sensor field may exist. For each data aggregation scheme, the resulting virtual topology (the set of LAs) is then fed into the ILP and heuristic algorithms. In both schemes, the algorithms will find the set of routes and MAs for the two-level (2L) aggregation scheme as well as the set of routes and aggregation points for the multi-level (ML) aggregation scheme as described earlier. The ILP problem is solved using the CPLEX linear programming package [41]. In the real problem, the ILP and the heuristics can be solved at the BS node. The set of routes and the aggregation points obtained for both schemes are used for further simulation experiments in order to evaluate the energy-delay tradeoffs as will be explained later in this section. We performed separate sets of experiments to investigate the impact of different parameters (all reported results are averaged over 10 runs). In particular, we studied the following performance issues:

Aggregation versus No-Aggregation: We consider the lifetime of the network without aggregation to be the baseline network lifetime, which is taken as 1. We also define the performance metric L as the ratio of the system lifetime achieved using aggregation to that obtained without using aggregation. We refer to L as the lifetime extension ratio. We performed separate sets of experiments for both 2L and ML aggregation schemes. The results are shown in Table 3 for the 2L scheme for different values of n and number of MAs M, and in Table 4 for the ML scheme for different values of n and number of groups χ.

As shown in Table 3, all schemes with aggregation result in prolonging the lifetime of the sensor network. The value of lifetime extension ratio (L) is the highest with the optimal approach, which can be as large as 5, and sometimes even larger. Out of the group of the approximate approaches, the LBA approach has the best results. However, the GA approach is not very far behind, which makes it a good candidate for use. Table 4 shows values for L for different values of LA nodes, n, and when the num-

Table 4
Lifetime extension ratio (L) for different values of n and χ in ML schemes.

n	χ	No aggregation	GA	CBAH	*ILP*
8	3	1	11.98	12.03	14.04
	4	1	10.64	11.13	12.01
10	3	1	12.36	12.24	15.34
	5	1	17.69	17.97	19.98
15	5	1	10.25	11.63	14.97
	6	1	21.22	22.15	24.57

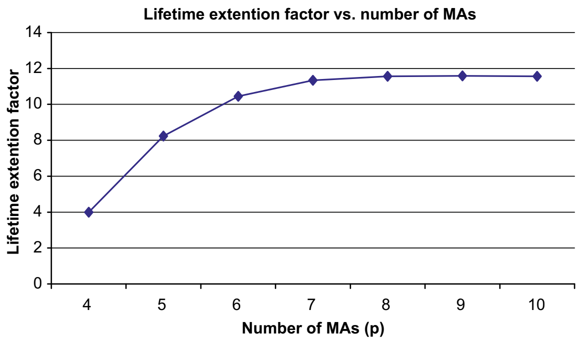


Fig. 7. The effect of increasing the number of MAs on the extension of the network lifetime when $n = 15$ under 2L aggregation scheme.

ber of groups, χ, is varied for every value of n with multiple levels of aggregation allowed. The value of L can be as large as 24 using the ILP especially with a greater number of groups, and sometimes even larger. As noted, the value of L for 2L scheme is lower than the ML scheme. This indicates that further levels of data aggregation can result in greater levels of power savings. From the results in Table 4, it is clear that the performance of CBAH and GA are not far away from the optimal performance obtained when the ILP is used. However, CBAH runs much faster than the ILP especially for large values of n.

Effect of number of MAs: we study the effect of varying the number of MAs, M, for a fixed value of n on the increase in the network lifetime in the 2L aggregation scheme. For this purpose, we fixed n to 15 nodes so that most of the network field is covered and connected. Then, we varied the values of M from 3 to 10 and measured the value of L. Fig. 7 shows the lifetime extension factor (L) versus the number of MAs. As shown in the figure, when the number of MAs increases, the lifetime is increased until $p = 7$ where no further improvement in the lifetime is obtained. In most cases, it was observed that increasing M beyond $\frac{n}{2}$ does not result in any significant increase in L.

Energy-Delay Tradeoffs Results: In this part, we measure the aggregation delay incurred due to the aggregation process for various schemes, as defined earlier. We studied the delay while varying the number of sensor nodes in the field. We set $r = 20$ meters and varied n from 6 to 15. M was set to $n/2$, and the number of groups was 3. Fig. 8 shows the aggregation delays as the number of LA nodes increases under the 2L and ML aggregation schemes, and as evaluated using exact, and heuristic approaches. The increase in the node density helps to fill the zones in the virtual architecture and increases the node density, and hence the connectivity of the virtual graph. Therefore, the aggregation delay decreases for the 2L aggregation as the number of hops to reach the second-level aggregation point decreases. However, the case is not true for ML aggregation, where many source nodes of the same group may share several aggregation points along the route to BS, and hence more aggregation delays can be experienced. It can therefore be concluded that if the sensor network is designed for time critical applications, two-level aggregation scheme would perform better on the expense of additional power consumption. If data gathering and reporting delay is not a concern, multi-level aggregation schemes would be a good choice in this case as these schemes consume less power and hence allow for longer network lifetime.

Comparison with other data aggregation schemes: In this part, we compare our two-level and multi-level routing with data aggregation schemes to some related work in the literature. In particular, we compare CBAH to Directed Diffusion (DD) framework [3], Pegasis [6], MLDA [29], and LEACH [5]. For comparisons with [3,5,6], we have uniformly distributed 100, 200, and 300 sensor nodes in a $200 \times 200\ m^2$ fixed sensor field. We set transmission range to 40 m. Hence, the sensor field has 25 zones (clusters) regardless of the number of nodes used. The number of groups in CBAH were set to 2, 4, and 6, respectively. To compare CBAH to [29], CMLDA was used since CBAH and CMLDA are both hierarchical. The same parameters and settings as described [29] were used, where a $100 \times 100\ m^2$ fixed sensor field was simulated with BS at (50, 300) m, 1 J/node, a packet size was fixed to 1000 bit data and 120 bit control packet. Also, the aggregation energy consumption was assumed to be 5nj/bit. We vary the number of groups in CBAH to be 3, 5, and 6 groups, respectively, where one event is reported by a subset of sensor nodes within each group. The sensor subsets are selected randomly. Our objective is to study the impact of different numbers of sources and different numbers of

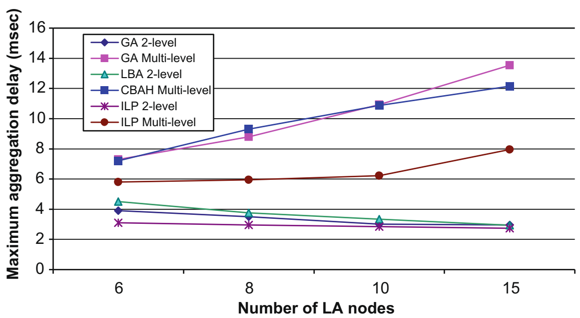


Fig. 8. Aggregation delays when two-and multi-level data aggregation are used in small networks; $\chi = 3$.

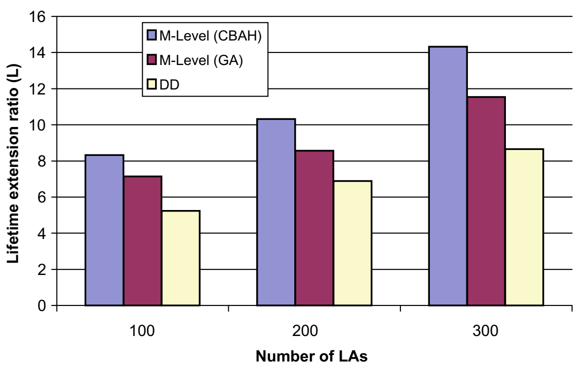


Fig. 9. Lifetime extension ratio (L) for our multi-level data aggregation schemes and Directed Diffusion (DD). M-level: multi-level.

groups on the network lifetime and the amount of aggregation delays experienced in the network. For the sake of comparison with these schemes, the duplicate suppression aggregation function is used. Notice also that DD does not need any upfront knowledge about network configuration and data sizes, nor does it require a centralized solution like other schemes.

Figs. 9 and 10 plot the lifetime extension ratio (L) and aggregation delays for our approximate schemes and the DD paradigm, respectively, as the number of LA source nodes increases. Our simulation results demonstrate that the clustering based heuristic (CBAH) can achieve larger increase in the lifetime of WSNs when compared to DD even that the algorithm is centralized. For a fixed number of LA source nodes and a fixed number of groups, our schemes consume less energy than DD, and hence extends the network lifetime. This is due to two reasons. First, query flooding in our scheme is confined to only horizontal and vertical directions, while in DD a query propagates throughout the whole network field before some paths are re-enforced. Second, our scheme searches for routes that balance the power consumption in the network, while DD makes no distinction between routes used that will carry data to the BS.

For other schemes, Table 5 shows a network lifetime comparison between CBAH and other schemes, namely, LEACH and PEGASIS for both cases when aggregation and no aggregation are used. Table 6 shows a network lifetime comparison between CBAH and CMLDA. In conclusion, our experimental results demonstrate that the CBAH can increase the system lifetime of large WSNs, when compared to the CMLDA. Note that both CBAH and MLDA (or CMLDA)

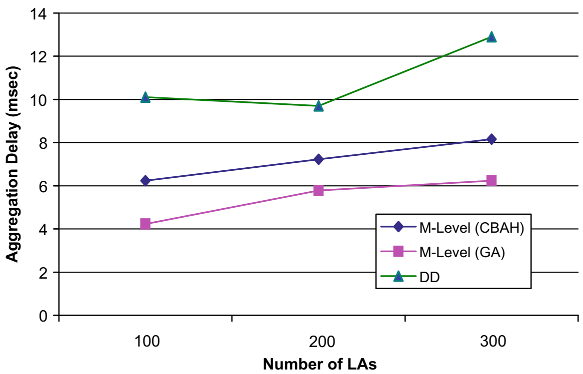


Fig. 10. Aggregation delays in both our multi-level data aggregation schemes and the Directed Diffusion scheme.

Table 5
Network lifetime for various schemes with aggregation and no aggregation used.

n	χ		CBAH	LEACH	PEGASIS
100	2	Aggregation	10994	6735	5373
100	2	No aggregation	2525	1549	2314
200	4	Aggregation	19544	6494	3798
200	4	No aggregation	2266	982	1339
300	6	Aggregation	27489	6158	4175
300	6	No aggregation	2014	964	2023

Table 6
Network lifetime in CBAH and MLDA.

n	Nodes/cluster	LAs	χ	CBAH	CMLDA
100	10	10	3	4233	3611
200	10	20	5	5648	4512
300	15	20	6	7568	5560
400	20	20	6	8995	6142
500	25	20	6	10439	6577

Table 7
Network aggregation delays in various schemes.

n	CBAH	LEACH	PEGASIS	CMLDA
100	2.0	2.10	2.60	2.4
200	3.6	2.74	4.35	3.91
300	4.5	3.80	5.96	5.43

are centralized where the BS is responsible of clusters formation and routes construction, i.e., algorithm implementation. Also, CBAH is called only once at the startup phase and the same settings (routes and clusters) will remain fixed for the whole network lifetime. However, the aggregated trees used in MLDA changes with time which may lead to synchronization and coordination problems between the sensor nodes. Hence, CBAH is much simpler than MLDA and requires much less time to converge making it more scalable.

We also collected the aggregation delays for various schemes as summarized in Table 7. For most cases, CBAH is able to minimize aggregation delays while at the same time largely enhances the network lifetime.

We also plot the data aggregation delay experienced in our heuristic schemes and in the DD (see Fig. 10) when the number of LAs increases. As shown in the figure, our schemes experience lower aggregation delays than DD. This is because in DD data forwarding paths from different sources may cross or overlap with each other anywhere in the network area, thus there is more interference when the number of sources is large, whereas in our scheme each LA source node sends data on the virtual grid, thus data flows on the grid faster and the routes are selected to balance power consumption in the network.

7. Conclusions

In this paper, we studied the maximum lifetime data gathering and routing problem in WSNs. We showed that cluster-based algorithms along with data aggregation and in-network processing can achieve significant energy savings in WSNs. This has a direct effect on prolonging the network lifetime. In particular, we developed GRASS (Grid-based Routing and Aggregator Selection Scheme), a scheme for WSNs that combines the ideas of fixed cluster-based routing together with application-specific data aggregation in order to enhance the wireless sensor network performance in terms of extending the network lifetime, while incurring acceptable levels of latency under data aggregation. Within GRASS, we have presented optimal as well as heuristic algorithms that solve the joint problem of optimal routing with data aggregation for the sake of maximizing the network lifetime. Our results show that, when compared to other approaches in the literature, the proposed scheme is able to improve the network lifetime while incurring acceptable levels of latency and without sacrificing quality. Hence, GRASS can attain the energy and latency efficiency needed for Wireless Sensor Networks.

Acknowledgements

This research was supported in part by the National Science Foundation under Grants CNS-0626822 and ECS-0601570. J.N. Al-Karaki was with Iowa State University. He is now with the Hashemite University, Zarqa 13115, Jordan. R. Ul-Mustafa was with Iowa State University. He is now with Microsoft, Redmond, WA 98052-6399, USA. A.E. Kamal is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA.

Appendix A. Proof of NP-completeness of RSP1

In this appendix, we prove the NP-completeness of the decision version of the *RSP*1 problem. We construct an instance of the *RSP*1 problem in which:

1. The power consumption to transmit a unit of data from any LA to any other LA is constant and the same for all LAs. This is taken as unity for simplicity.
2. That no aggregation is performed, i.e., the aggregation ratio is 1.
3. That $M = 2$, and that we have already solved the MA selection problem by choosing LA_j and LA_k, where the power (per byte) required to deliver data from the selected MAs to the BS is w^*, and from LA_l to the BS is w_l, such that

$$w * \sum_{i \in S} m_i < w_l m_i \quad \text{for } l \neq j,k, \quad \forall i.$$

That is, delivering all data generated in the network from either of these two MAs to the BS requires an amount of power which is less than delivering data from a single traffic stream from any other LA to the BS.

The decision version of *RSP*1 in this case would be: *Is there an assignment of the LAs to the two MAs such that the power consumption at both MAs is the same?*

Theorem 1. *Problem RSP*1 *is NP-complete*

Proof. Given a certificate which corresponds to a solution of the *RSP*1 decision problem, it can be verified in polynomial time whether this solution will satisfy the power consumption requirement by straight forward substitution.

Now, we show that RSP1 is NP-hard by reducing the *set partitioning* problem, which is an NP-complete problem [39], to the constructed instance of RSP1 given above in polynomial time. That is, we show that

$$\text{set partitioning} \leqslant {}_p RSP1.$$

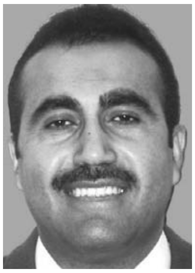
Given an instance of the set partitioning problem represented by the set, $S = \{s_i : 1 \leqslant i \leqslant N, s_i > 0, s_i \in \mathbf{Z}^+\}$, we would like to partition S into two subsets S_1 and S_2, such that $S = S_1 \bigcup S_2$, $S_1 \bigcap S_2 = \phi$, and $\sum_{s_i \in S_1} s_i = \sum_{s_i \in S_2} s_i$. We map each element s_i to the amount of traffic m_i generated by LA_i. This mapping can be done in polynomial time.

Therefore, if there is a YES answer to the constructed instance of the *RSP*1 problem, there will be a YES answer to the set partitioning problem, and this will solve the set partitioning problem. Therefore, RSP1 is NP-hard. The above two parts prove that RSP1 is NP-complete. □

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[2] J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Wireless Communications 11 (6) (2004) 6–28. Dec..
[3] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of ACM MobiCom'00, Boston, MA, 2000, pp. 56–67.
[4] B. Krishnamachari, D. Estrin, S. Wicker, Impact of data aggregation in wireless sensor networks, in: International Workshop on Distributed Event-Based Systems, Vienna, Austria, July 2002.
[5] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, in: Proceedings of HICSS'00, January 2000.
[6] S. Lindsey, C. Raghavendra, PEGASIS: power efficient gathering in sensor information systems, in: IEEE Aerospace Conference Proceedings, vol. 3, 2002, pp. 1125–1130.
[7] A. Manjeshwar, D.P. Agarwal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in: First International Workshop on Parallel and Distributed Computing-Issues in Wireless Networks and Mobile Computing, April 2001.
[8] A. Manjeshwar, D.P. Agarwal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in: Proceedings of IEEE IPDPS 2002, pp. 195–202.
[9] V. Rodoplu, T.H. Meng, Minimum energy mobile wireless networks, IEEE Journal Selected Areas in Communications 17 (8) (1999) 1333–1344.
[10] S. Bandyopadhyay, E. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of INFOCOM 2003, vol. 3, pp. 1713–1723.
[11] L. Li, J.Y. Halpern, Minimum energy mobile wireless networks revisited, in: IEEE International Conference on Communications (ICC) 2001, vol. 1, pp. 278–283.
[12] Q. Li, J. Aslam, D. Rus, Hierarchical power aware routing in sensor networks, in: Proceedings of the DIMACS Workshop on Pervasive Networking, May, 2001.
[13] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, A two-tier data dissemination model for large-scale wireless sensor networks, in: Proceedings of ACM MobiCom, 2002.
[14] W. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: Proceedings of ACM Mobicom, 1999, pp. 174–185.
[15] J. Kulik, W.R. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, Wireless Networks 8 (2002) 169–185.
[16] Y. Yu, D. Estrin, R. Govindan, Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks, UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.
[17] J.H. Chang, L. Tassiulas, Maximum lifetime routing in wireless sensor networks, IEEE/ACM Transaction on Networking 12 (4) (2004) 609–619. Aug..
[18] K. Kalpakis, K. Dasgupta, P. Namjoshi, Maximum lifetime data gathering and aggregation in wireless sensor networks, in: Proceedings of IEEE Networks, 2002.
[19] C. Buragohain, D. Agrawal, S. Suri, Power aware routing for sensor databases, in: Proceedings of IEEE Infocom, 2005.
[20] H. Gupta, V. Navda, S. Das, V. Chowdhary, Efficient gathering of correlated data in sensor networks, in: Proceedings of ACM MobiHoc'05, pp. 402–413.
[21] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, Wireless Networks 8 (5) (2002) 481–494.

[22] J.N. Al-Karaki, R. Ul-Mustafa, A.E. Kamal, Data aggregation in wireless sensor networks: exact and approximate algorithms, in: Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR), Phoenix, Arizona, USA, April 2004, pp. 18–21.
[23] S. Madden, M. Franklin, J. Hellerstein, W. Hong, TAG: a tiny aggregation service for ad hoc sensor networks, OSDI 2002, Boston MA, December 2002.
[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.
[25] S. Singh, C.S. Raghavendra, PAMAS: power aware multi-access protocol with signalling for ad hoc networks, ACM Computer Communication Review 28 (3) (1998) 5–26.
[26] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of INFOCOM 2002, pp. 1567–1576.
[27] J.N. Al-Karaki, A.E. Kamal, Efficient virtual-backbone routing in mobile ad hoc networks, Elsevier Computer Networks 52 (2) (2008) 327–350.
[28] A. Giridhar, P.R. Kumar, Maximizing the functional lifetime of sensor networks, in: Proceedings of the Information Processing In Sensor Networks (IPSN'04).
[29] K. Kalpakis et al, Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks, Computer Networks Journal 42 (6) (2003).
[30] K. Dasgupta et al., Improving the lifetime of sensor networks via intelligent selection of data aggregation trees. in: Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'03), 2003.
[31] M. Esseghir, N. Bouabdallah, G. Pujolle, Energy provisioning model for maximizing wireless sensor network lifetime, in: Global Information Infrastructure Symposium, 2–6 July 2007, pp. 80–84.
[32] J. Hung-Chin, H. Lee, Efficient energy management to prolong wireless sensor network lifetime, in: Third IEEE/IFIP International Conference in Central Asia on Internet, 26–28 September 2007, pp. 1–4.
[33] F. Sun, M. Shayman, Prolonging network lifetime via partially controlled node deployment and adaptive data propagation in WSN, in: Annual Conference on Information Sciences and Systems, 14–16 March 2007, pp. 226-231.
[34] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, Z.M. Wang, A new MILP formulation and distributed protocols for wireless sensor networks lifetime maximization, in: IEEE International Conference on Communications, vol. 8, June 2006, pp. 3517–3524.
[35] H. Luo, Y. Liu, S.K. Das, Routing correlated data in wireless sensor networks: a survey, IEEE Network 21 (6) (2007) 41–47.
[36] J. Chen, L. Gong, P. Zeng, Y. Yang, Several repacking algorithms for data aggregation in wireless sensor network, in: IEEE International Conference on Information Acquisition, 2006, pp. 632–637.
[37] K. Lu, L. Huang, Y. Wan, H. Xu, Energy-efficient data gathering in large wireless sensor networks, in: Second International Conference on Embedded Software and Systems, 2005, pp. 5–10.
[38] J.N. Al-Karaki, Infrastructureless wireless networks: cluster-based architectures and protocols, PhD Thesis, Electrical and Computer Engineering, Iowa State University, 2004.
[39] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[40] The ns-2 simulator. <http://www.isi.edu/nsnam/ns>.
[41] The CPLEX linear optimization package, V8.1.0. <http://www.ilog.com/products/cplex>.



Jamal N. Al-Karaki is currently an Assistant Professor at the Department of Computer Engineering, The Hashemite University, Zarka, Jordan. He earned his Ph.D. from Iowa State University in 2004 with a research excellence award. His main research interests include distributed systems, wireless networking, and Mobile Internet. He has published over 25 articles in these subjects. He is also the recipient of the best paper award from IEEE ISCC 2006.

Raza Ul-Mustafa is now with Microsoft Corporation. He completed his Ph.D. in Computer Engineering at the Department of Electrical and Computer Engineering, Iowa State University, USA, in 2004. Earlier, he received his M.S. from Kuwait University, Kuwait, in 1999 and B.Sc. from University of Azad Jammu & Kashmir, Pakistan, in 1995, both in Electrical Engineering.

He also worked with Lucent Technologies and Paktel. His research interests include WDM networks, wireless sensor networks and high speed networks. He is a member of IEEE and Tau Beta Pi.

Ahmed E. Kamal received a B.Sc. (distinction with honors) and an M.Sc. both from Cairo University, Egypt, and an M.A.Sc. and a Ph.D. both from the University of Toronto, Canada, all in Electrical Engineering in 1978, 1980, 1982 and 1986, respectively. He is currently a professor of Electrical and Computer Engineering at Iowa State University. Earlier he held faculty positions in the Department of Computing Science at the University of Alberta, Canada, and the Department of Computer Engineering at Kuwait University, Kuwait.

He was also an adjunct professor at the Telecommunications Research Labs, Edmonton, Alberta.

His research interests include high-performance networks, optical networks, wireless and sensor networks and performance evaluation. He is a senior member of the IEEE, a member of the Association of Computing Machinery, and a registered professional engineer.

He was the co-recipient of the 1993 IEE Hartree Premium for papers published in Computers and Control in IEE Proceedings for his paper entitled Study of the Behaviour of Hubnet.

He served on the technical program committees of numerous conferences and workshops, was the organizer and co-chair of the first and second Workshops on Traffic Grooming 2004 and 2005, respectively, was the co-chair of the Technical Program Committees of the Optical Symposium of Broadnets 2006, the Communication Networks and Services Research (CNSR) conferences 2006, and the Optical Networks Symposium of the IEEE Globecom 2007, and the Technical Program Chair of the IEEE/ACS AICCSA 2008 conference.

He is an area editor of the Computer Networks journal, published by Elsevier.