

Energy Balance Based Lifetime Maximization in Wireless Sensor Networks Employing Joint Routing and Asynchronous Duty Cycle Scheduling Techniques

Mohsen Kariman-Khorasani · Mohammad Ali Pourmina · Ahmad Salahi

Published online: 1 March 2015
© Springer Science+Business Media New York 2015

Abstract This paper introduces a novel energy balance based lifetime maximization issue in wireless sensor networks employing joint routing and asynchronous duty cycle scheduling techniques titled as EB-JRADCS problem. To formulate the EB-JRADCS problem a new asynchronous MAC protocol utilizing flooding of RTS and random sending of CTS named FRTS–RCTS is proposed. This protocol leads to new constraints called flow sharing (FS) constraints that joint the network lifetime maximization parameters including flow rate of information on any route and duty cycle of nodes. It is shown that the modeled EB-JRADCS problem can be expressed as a signomial geometric programming problem. Due to the complexity of solving the achieved problem, first it is converted into a simpler problem by relaxing FS constraints from equal to unequal form. Then the simplified problem is solved with the aid of a specific convexification method and the global optimum solution of the network lifetime is evaluated under various scenarios. The achieved optimum solution can be used as a benchmark for evaluating and comparing distributed and heuristic methods that aim to extend the network lifetime.

Keywords Duty cycle scheduling · Flow sharing (FS) constraints · Maximum lifetime routing · Signomial geometric programming (SGP)

M. Kariman-Khorasani (✉) · M. A. Pourmina
Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
e-mail: mohsen.kariman@gmail.com

M. A. Pourmina
e-mail: pourmina@srbiau.ac.ir

A. Salahi
Information and Communication Security Department, Research Institute for ICT (ex ITRC), Tehran, Iran
e-mail: ahmadsalahi@yahoo.com

1 Introduction

A wireless sensor network (WSN) is consisted of very tiny sensor nodes which are spread out in a specific geographical area. This network is usually used in monitoring or surveillance applications. Each sensor node collects information from its surrounding environment and forwards it to one or several sink nodes in a single or multi hop way. In such networks, each of the sensor nodes is powered by a very tiny battery and usually due to utilizing of these networks in hazardous or out of reach environments, replacing node batteries would be rarely possible. So the main problem in these networks is that sensor nodes can operate for a long duration of time by such limited batteries. This issue has guided current studies toward methods to increase or, in optimum case, maximize the network lifetime in WSNs. Since the definition of network lifetime will completely depend on the network application, often there is no unified definition for the lifetime of a sensor network. In most of the studies, the lifetime of a sensor network is considered the time duration in which the first node of the network is de-energized [1, 2]. In some others, the network lifetime is considered the time duration in which a certain percentage of network nodes are deactivated [3]. In other studies, lifetime network is defined the time duration in which the coverage area of sensor network for gathering and reporting data is reduced to the given threshold level [4]. In a common sensor node, the battery energy is consumed by different parts of the sensor node. But the most energy consumed is related to the communicational subsystem of each node [5]. In this subsystem, four processes including sending data, receiving data, idle listening (IL) for receiving data and sleeping are done. These activities will be effective on the other energy wasting factors like collision and overhearing. Thus the way of performing the above quadruplet activities will be intensely influential on the lifetime of each node and hence overall network lifetime. Based on this fact, lifetime extension techniques in sensor networks can be divided into two main branches: energy efficient routing methods and duty cycle scheduling techniques.

In the first branch, the proper assignment of different routes of network for sending and receiving data, and in the second branch suitable timing mechanism related to the processes of IL and sleeping are considered in order to access the most efficiency of energy. Many methods have been suggested for lifetime maximization of sensor networks based on the energy efficient routing methods which are common as maximum lifetime routing problem. The issue related to maximum lifetime routing problem has been pioneered by Chang and Tassiulas [1] and modeled as a mathematical linear programming problem. In this method, authors have used a centralized modeling method based on calculating the flow rate of data on the connecting links of sensor nodes as independent optimizing parameters. They have assumed the network lifetime as the time duration in which the first node of the network runs out of energy. Madan and Lall [6] were the first researchers who presented a pure distributed classic solution to this problem. In both of methods presented in [1, 6], it has been assumed that all of the data communication among the nodes is performed in multi hop way and there is no direct link between sensor nodes and the sink nodes. But this assumption will cause lack of energy balance in the network nodes and usually the closer nodes to the sink deactivate sooner than the other ones. Thus the network lifetime is often limited to the lifetime of closer nodes to the sink. To prevent this deficiency, in [3] the issue of lifetime maximization, based on energy balance has been presented by utilizing both single and multi hop transmission policies. This issue has been formulated by an integer linear programming problem and also a distributed protocol named distributed energy balance routing (DEBR) has been presented to practical implementing the scheme. In DEBR protocol, it has been allowed to each node to choose its

route toward the sink node by a neighboring node as a data relay node or directly. This choice has been performed based on a specific criterion of the energy cost.

In the second branch of researches, increasing lifetime of the sensor network has been carried out by duty cycle scheduling techniques. Duty cycle scheduling is defined as determining time durations of waking and sleeping of a sensor node. The foundation of all methods of extending the lifetime based on duty cycle scheduling, is reducing the time of activity of a sensor node or in other words, increasing the time of its sleeping; when neither node receives any data nor sends it. In fact, at this time sensor node waiting for receiving data ought to keep its receiver on and listen to the radio channel. This situation is well known as IL mode. But this procedure will cause a lot of energy wastage in a sensor node. Because the experiments have shown that the amount of energy consumed in IL mode, is nearly equal to energy consumed in receiving data mode [5]. The duty cycle scheduling techniques can be divided into two main categories, including synchronous and asynchronous methods [5]. In synchronous methods, it is assumed that sensor nodes in their communicational routes are aware of their sleeping and waking time durations of each other. Synchronizing the network nodes especially in large scales imposes heavy signaling load on the network. On the contrary in asynchronous techniques, duty cycle scheduling of nodes is performed independent of each other; so handling these methods is easier to implement than synchronous ones. However they may be positioned in lower level in terms of energy efficiency and lifetime. Anastasi et al. [7] have suggested a new synchronous protocol for duty cycle scheduling named adaptive staggered sleep protocol (ASLEEP) in order to extend the lifetime of a data gathering tree based WSN. ASLEEP adjusts dynamically the sleep time duration of nodes based on some specific network parameters like traffic rate. In [8] a synchronous MAC protocol named S-MAC designed specifically for a sensor network has been suggested to improve the efficiency of traditional CSMA/CA MAC protocol. In S-MAC to reduce the IL time, periodical listening and sleeping has been used for duty cycle scheduling of each sensor node. In S-MAC protocol, in order to synchronizing the nodes, they exchange a certain packet named SYNC. In [9] an asynchronous MAC protocol called B-MAC using a preamble along data packet has been suggested for reducing IL time duration of nodes. In [10] the issue of asynchronous duty cycle scheduling has been presented and formulated with two main objectives including lifetime maximization and overall power consumption minimization.

In each of the aforementioned researches, extending or maximizing the network lifetime is done by focusing on any of the energy efficient routing or duty cycle scheduling branches; while considering both of these techniques jointly can be numbered as prosperous method to extend or maximize the network lifetime. Kartal et al. [11] have presented a lifetime maximization issue based on joint routing and duty cycle scheduling in WSNs and modeled the issue as a linear programming problem. In that research it has been assumed that there exists an underlying synchronized duty cycle scheduling mechanism. Also it has been supposed that all of the sensor nodes operate with the same duty cycle, but as the authors have mentioned, it is obvious that this assumption cannot guarantee the maximum lifetime in the network. In [12] also an issue named joint routing and sleep scheduling problem for lifetime maximization in sensor network has been presented. In [12], it has been assumed that sensor nodes are independent of each other in setting their sleeping time duration. Despite the advantages of both methods presented in [11, 12], they have a main drawback. In the both presented issues, it has been assumed that data transmitting of all nodes toward the sink is only performed in multi hop way. But this policy will cause lack of energy balance in the nodes and early deactivating of closer nodes to the sink.

In this paper, a novel energy balance based lifetime maximization issue in WSNs employing joint routing and asynchronous duty cycle scheduling techniques is introduced which

is called as EB-JRADCS problem. To model the EB-JRADCS problem a new asynchronous MAC protocol utilizing flooding of RTS and random sending of CTS named FRTS–RCTS is proposed. This protocol leads to new constraints called flow sharing (FS) constraints that joint the network lifetime maximization parameters including flow rate of information on any route and duty cycle of nodes. In fact in EB-JRADCS issue, three approaches will be considered. First, duty cycle scheduling of each node will be done in an asynchronous manner. Second, routing and duty cycle scheduling mechanisms are performed jointly. It means both of them are considered as the factors of lifetime maximization problem simultaneously. Third, lifetime maximization based on energy balance will occur in the network; so for this sake, sensor nodes are able to send their data into the sink nodes either in single hop or multi hop way. In this paper, first we show that EB-JRADCS problem can be expressed as a standard signomial geometric programming (SGP) problem. Due to the complexity of solving the achieved problem, first it is converted into a simpler problem by relaxing FS constraints from equal to unequal form. Also because of inherent non-convexity of the problem, we use a convexification method which has been presented in recent literature [13] in order to solve the problem and calculate overall optimum network lifetime.

The rest of the paper is organized as follows. In Sect. 2 after introducing the system model and assumptions, we will illustrate the new asynchronous FRTS–RCTS protocol and resulted FS constraints in detail. Then we compute energy constraints according to precise computations of energy consumption in each node based on proposed protocol and subsequently describe the way of formulating EB-JRADCS problem based on all of the obtained constraints. In Sect. 3 by defining a SGP problem, we will show that EB-JRADCS problem is easily convertible to a SGP problem and then we will briefly present the procedure of solving the problem. Also in this section, we will show that the problem can be converted into a simpler problem by relaxing FS constraints from equal to unequal form. In Sect. 4 simulation results including maximum network lifetime under various scenarios, and also flow rate of information and duty cycle of nodes on a given scenario will be presented. Eventually in Sect. 5 we will conclude the paper.

2 System Model, Assumptions and EB-JRADCS Problem Formulation

Consider a WSN where V is the set of all sensor nodes. N sensor nodes are uniformly distributed and fixed in a specific region. Also, M sink nodes are placed in different locations of this area. Each sensor node contains primary battery energy E_i and generates fixed data rate of g_i . Also it has been assumed that sensor nodes have ability of adjusting their transmission power level based on their own distance to the target node. Each sensor node forwards its generated information, either directly to the sink or to one of its neighbors as a data relaying node. In the network consisted of multiple sinks, in the condition of directly sending of information to the sink, data is sent to the nearest sink. To avoid data collision in directly sending case, we assume that sending data is done on a separate channel. It is clear that the number of neighbors of each node will completely depend on its transmitting power level, as by increasing power transmission level of each node, more nodes as adjacent nodes or in other words relaying nodes will be covered. Now consider that each node covers communicational radius R by sending specific power level in its neighborhood. Later we will investigate the effect of changing radius R or equivalently changing power transmission level on the network lifetime. The set of neighbors of a sensor node like node i in coverage radius R is shown by N_i^R . If r_{ij} denotes

the rate of transmitting data from node i to j and S_i represents the nearest sink to node i , then flow conservation constraints will be shown as follows:

$$\sum_{j \in N_i^R + \{S_i\}} r_{ij} = \sum_{k: i \in N_k^R} r_{ki} + g_i, \quad \forall i \in V \quad (1)$$

Equation 1 shows that the sum of outgoing data rate from any node involving outgoing data rate toward neighbors and outgoing data rate directly to the sink node is equal to incoming data rate from nodes which node i is their neighbor plus data generated by node i itself. In fact, since all generated data should be sent to the sink nodes, the incoming and outgoing data rate at a sensor is the same.

2.1 FRTS–RCTS Protocol and Flow Sharing (FS) Constraints

In this section, by designing a new asynchronous MAC protocol named FRTS–RCTS, it is shown that lifetime maximization parameters, including flow rate of information on any route and duty cycle of nodes are jointed via new constraints named FS constraints. In traditional asynchronous MAC protocol called CSMA/CA, duty cycle scheduling in order to reduce energy consumption is not supported. Hence, each sensor node in order to make a connection with another node, just forwards RTS packet toward it. Then to overcome hidden terminal problem the receiver node sends CTS packet in response to the transmitter node. To support duty cycle scheduling in the proposed protocol, the procedure of CSMA/CA protocol is modified in both stages of RTS and CTS transmissions. It should be noted that the fundamental idea in designing the new protocol is independence of duty cycle of nodes from each other. To this end, first a clear definition of duty cycle should be presented. Duty cycle of a node can be defined as the ratio of its activity time duration to a given fixed time duration called duty cycle period. In each duty cycle period a sensor node performs some operations like data sending, data receiving and IL in its waking time and during the remaining time of the period goes to sleep mode for saving its battery energy. In the sleep mode, the sensor node deactivates all of its communicational equipment whether transmitter or receiver, so that the energy consumption in this situation would be negligible. Since in WSNs due to low traffic rate, the time relating to sending and receiving data in comparison to the time of IL is too small, usually the duty cycle of a node can be considered as the ratio of IL time to the duty cycle period. In this paper we utilize this principle for calculating of node duty cycle. By considering this assumption, in the following we will illustrate the procedure of FRTS–RCTS protocol in detail.

If a sensor node like j , the neighbor of node i , works with a given duty cycle denoted by dc_j and node i in an arbitrary moment, sends a RTS packet to this node, the probability of that node j be in IL mode is dc_j . Now when node i , needs to send a data packet to the sink node, due to the lack of awareness of its waking neighbors, first it sends the RTS packet toward its neighbors in flooding way in a given radius R . The set of neighbors of sensor node i in coverage radius R is shown by N_i^R and the number of them is denoted by $|N_i^R|$. Also the set of waking and sleeping neighbors of node i is represented by WN_i^R and SN_i^R respectively. Now in terms of waking or sleeping modes of neighbors of node i , FRTS–RCTS protocol works exactly after sending RTS, as one of the two following different cases:

2.1.1 FRTS–RCTS Protocol: Case 1

Among all neighbors of node i , k neighbors are awake or in other words in IL mode. In this case, all of these neighbors receive sent RTS packet from node i . In this condition any of these neighbors can send CTS packet in response to the received RTS packet. But by sending CTS packet immediately after receiving RTS, by at least two neighbors, collision will be occurred. To overcome this obstacle, FRTS–RCTS protocol acts in two main phases which is called CTS random scheduling (CRS) phase and data scheduling (DS) phase. In CRS phase, the transmitter node i , forms a time slotted window consisted of $|N_i^R|$ slots. In fact each of the slots is considered for receiving one of the sending CTS packets. Thus the time duration of each slot in this window is equal to the time duration needed to receive one CTS packet. The transmitter node along RTS packet sends a random and uniform sequence of numbers between 1 and $|N_i^R|$ such that each of the numbers corresponds to one of the neighbors. In other words to each of the neighbors whether awake or asleep, one random number is assigned. In fact, each random number specifies a slot number for receiving CTS packets from the neighbors. Each waking neighbor that has received the RTS packet ought to send CTS packet in its specified slot number. To prevent energy wastage, each waking neighbor goes to sleep mode upon receiving RTS packet and its own slot number. Then it awakes in the assigned time slot and sends its CTS packet and goes to sleep mode again until the time of receiving data that will be happened in DS phase. In contrast, the RTS transmitter means node i , listens to the radio channel until receiving the first CTS packet from its neighbors.

When the first CTS packet is received from a neighbor in n th slot of CRS phase, to avoid collision between data packet and probably the remaining CTS packets from other waking neighbors and also minimize the energy wastage in node i , this node goes to sleep mode and postpones sending its data packet at least as much as $|N_i^R| - n$ time slots. After $|N_i^R|$ time slots, CRS phase is finished. After the end of CRS phase, DS phase is started. In this phase a time slotted window consisted of $|N_i^R|$ slots is used similarly. Because if sending data is done immediately after CRS phase, then data overhearing will be accrued by other senders of CTS. This operation causes extra energy wastage in the CTS senders, because other senders of CTS

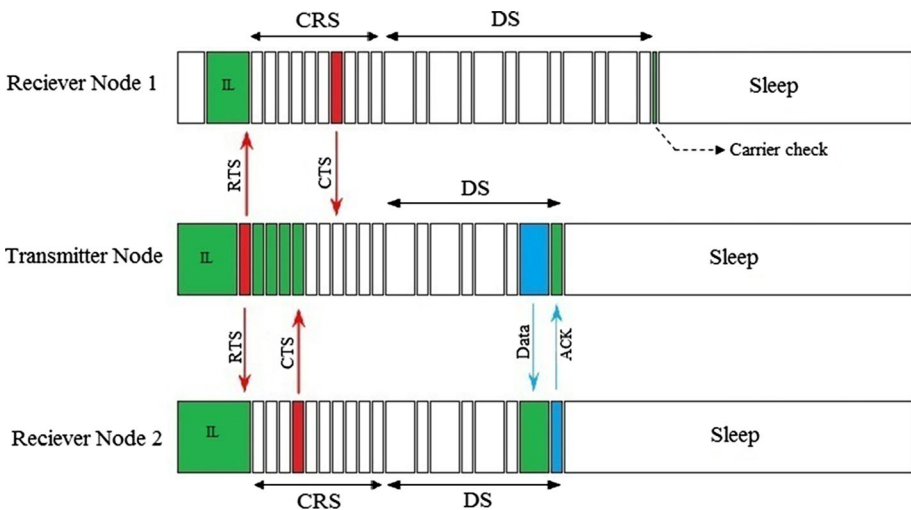


Fig. 1 FRTS–RCTS protocol timing (case 1)

packet are waiting to receive data packet too. Each slot is consisted of two parts in this phase. The first part is used for sending data packet from transmitter to receiver and the second part is used for sending ACK packet from receiver to transmitter conversely. In DS phase, if data transmitter node has been received the first CTS packet by a node like j in n th slot of CRS phase, so it will send data packet in n th slot of DS phase for node j too. To minimize energy wastage in this phase also, the transmitter node just awakes in slot of sending data. Also in this phase, overhearing never occurs by any other CTS senders, because all CTS senders awake only in the same slot of DS phase that is corresponding to their own specified CTS slot. In fact, the node which has sent its CTS packet later ought to check available carrier of the channel in its own data time slot only. In this situation, if the carrier is not heard, the node goes back to sleep mode again. In Fig. 1, the timing related to FRTS–RCTS protocol in the case 1 has been shown. In this figure white color is used for showing node sleeping mode. According to the Fig. 1, it has been assumed that transmitter node has ten neighbors and at the moment of RTS sending, only two neighbors, means neighbors 1 and 2 are active or in idle listing mode. Also it is supposed that assigned CTS random slots are the 7th and 4th slots for node 1 and 2 respectively which has been determined by node i . Therefore these nodes send CTS packet in the given time slots and the node 2 which has sent CTS packet sooner than node 1, will be chosen as data receiver node. Hence after the end of CRS phase, data packet will be sent to the chosen node means node 2 in the 4th slot of DS phase. This node also, awakes in the 4th time slot of DS phase and receives the sent data packet. Also node 1, which has sent CTS packet in 7th slot, awakes to receive the data packet in 7th slot of DS phase; but by checking the channel and lack of carrier existence, goes to sleep mode again. As it is seen in the Fig. 1, to prevent energy wastage of transmitter node, it goes to sleep mode immediately after receiving the first CTS packet in CRS phase until the moment of data transmitting in DS phase.

Now based on the illustrated FRTS–RCTS protocol procedure, we can calculate the probability of receiving a data packet by one of the neighbors of node i like j . If the set of waking neighbors of node i , including j in radius R is denoted by WN_i^j and the number of them are represented by $|WN_i^j|$, then the probability of receiving a data packet by node j is:

$$p(r = j | WN_i^j) = \frac{1}{|WN_i^j|}, \quad WN_i^j \subseteq N_i^R \tag{2}$$

In fact due to the random and uniform assignment of slot numbers to all neighbors, the probability of sending the first CTS packet and consequently, data receiving by any of the neighbors will be the same. But due to independence of duty cycle of nodes the probability that the set of waking nodes including node j be WN_i^j , would be calculated as follows:

$$p(WN_i^j) = \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k), \quad WN_i^j \cup SN_i^j = N_i^R \tag{3}$$

In (3), SN_i^R represents the set of sleeping neighbors of node i . Finally, by using total probability and conditional probability law, the probability of receiving a data packet by one of the neighbors of node i like j denoted by $p(r = j)$ is:

$$\begin{aligned} p(r = j) &= \sum_{WN_i^j \subseteq N_i^R} p(r = j | WN_i^j) p(WN_i^j) \\ &= \sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \end{aligned} \tag{4}$$

In fact in (4) the summation operator is done on the all of the possible subsets of N_i^R including j .

2.1.2 FRTS–RCTS Protocol: Case 2

All of the neighbors of node i are in sleep mode. In fact in this case, WN_i^R is an empty set or in other words SN_i^j is equal to N_i^R . In this situation, node i after sending RTS packet and waiting duration of time equal to $|N_i^R|$ time slots, by increasing its transmitter power level according to its distance to the nearest sink and switching to the second channel, after sending RTS packet to the sink node directly and receiving the corresponding CTS packet, sends data packet to the sink node. In Fig. 2 the protocol timing related to this case have been shown. The probability of receiving a data packet by the sink node that is sent by node i denoted by $p(r = S_i)$ is:

$$p(r = S_i) = p(WN_i^j = \emptyset) = p(SN_i^j = N_i^R) = \prod_{k \in N_i^R} (1 - dc_k) \tag{5}$$

In (5), S_i represents the nearest sink to node i . Now according to the two possible cases in the proposed protocol, in sending each data packet by transmitter node i , either one of the members of N_i^R like j with probability of $p(r = j)$ would be as a data relay node or the nearest sink with probability of $p(r = S_i)$ would be as the receiver. Given that the sum of all the probabilities in a random event is equal to 1, we have:

$$\left[\sum_{j \subseteq N_i^R} \sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \right] + \prod_{k \in N_i^R} (1 - dc_k) = 1 \tag{6}$$

Equation 6 will be used in Sect. 3 to simplify the EB-JRDCS problem.

Now consider that the network, works for a long duration of time T. If during this time, node i sends R_i data packets, then the number of sent packets for all of the neighbors and also the sink node will be written as follows:

$$R_i = \sum_{j \in N_i^R} R_{ij} + R_{iS_i} \tag{7}$$

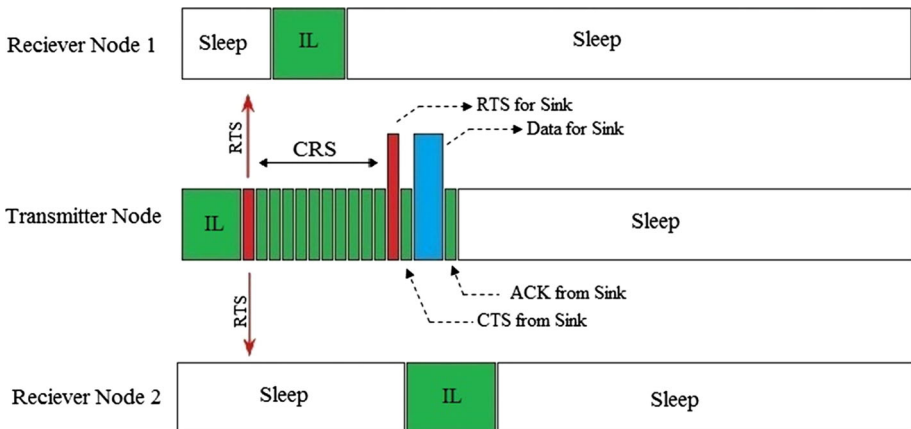


Fig. 2 FRTS–RCTS protocol timing (case 2)

In (7) R_{ij} is all of the outgoing packets from node i to j and R_{iS_i} is all of outgoing packets from node i to the sink S_i . So it can be said that the ratio of all the outgoing packets from sensor node i to j , to the total number of outgoing packets from node i , is equal to the probability of successful receiving data by node j and similarly, the ratio of all the outgoing packets from node i to the sink S_i , to the total number of outgoing packets from node i , is equal to probability of successful receiving data by the sink S_i , so we can write:

$$\frac{R_{ij}}{R_i} = p(r = j) \tag{8}$$

$$\frac{R_{iS_i}}{R_i} = p(r = S_i) \tag{9}$$

If sending all packets by node i , happens in the time duration of T , by dividing numerator and denominator of fractions in the left side hand of (8) and (9) to T and using (4) and (5) we have:

$$r_{ij} = \left[\sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \tag{10}$$

$$r_{iS_i} = \left[\prod_{k \in N_i^R} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \tag{11}$$

In the above equations or constraints in other words, the share of outgoing data flow rate to any of neighbors or to the sink node, in proportion to the total amount of outgoing data flow rate by node i , is calculated. So that's why we have named these equations as FS constraints. Also as it is seen, data flow rates on any route and duty cycle of nodes are tightly jointed by these constraints.

2.2 Energy Consumption Model

In each sensor node, data sending, data receiving and IL operations are the major factors of energy consumption. As other operations like sensing or information processing can be considered as negligible factors from this point of view. We use the basic model which is presented in [14] for energy consumption modeling in sending and receiving data statuses. According to this model, the energy required to send one unit of data from node i to j can be determined as follows:

$$E_t^{ij} = e_{elec} + e_{amp}(d_{ij})^\beta \tag{12}$$

where e_{elec} and e_{amp} are fixed coefficients depending on energy consumption of electronic equipment and transmitter amplifier module of sensor node respectively. Also d_{ij} is the Euclidean distance between node i and j and β is the path loss exponent which is typically in the range of 2–6, depending on the environment. In the case of receiving data, energy consumption of each node is fixed as follows:

$$E_r = e_{elec} \tag{13}$$

Finally in IL mode the amount of energy consumption is dependent on the radio module of sensor node. In respect to nothingness of energy wastage in sleep mode, it has been

ignored in this paper. According to the above mentioned model, in the next part we will compute the amount of energy consumption based on the proposed FRTS–RCTS protocol.

2.3 Energy Consumption Computation in FRTS–RCTS Protocol

Based on the Fig. 1, the energy consumed by each node like i , in sending one data packet to a neighbor node like j , is consisted of energy consumption of sending one RTS packet in radius R ($E_{t_RTS}^R$), energy consumption of waiting time to receive the first CTS packet from a neighbor ($E_{w_CTS}^i$), energy consumption of receiving one CTS packet (E_{r_CTS}), energy consumption of sending one data packet to node j ($E_{t_DATA}^{ij}$) and finally energy consumption to receive the ACK packet (E_{r_ACK}). So the energy consumption of node i , to send a data packet to a neighbor like j , without considering energy consumed when being in IL mode to receive the first CTS packet denoted by E_t^{ij} will be calculated as follows:

$$E_t^{ij} = E_{t_RTS}^R + E_{r_CTS} + E_{t_DATA}^{ij} + E_{r_ACK} \tag{14}$$

For simplicity, energy consumed during IL mode to receive the first CTS packet will be computed separately. According to FRTS–RCTS protocol in case 2, if any CTS packet is not received after the end of CRS phase, transmitter node by adjusting its transmitting power level sends RTS packet to the nearest sink. In this situation, the energy consumption of node i , to send a data packet to the sink, without considering energy consumed when it is waiting to receive the first CTS packet, denoted by $E_t^{iS_i}$ will be calculated as follows:

$$E_t^{iS_i} = E_{t_RTS}^R + E_{t_RTS}^{iS_i} + E_{r_CTS} + E_{t_DATA}^{iS_i} + E_{r_ACK} \tag{15}$$

In (15), the energy consumption of sending RTS packet and sending data packet from node i to sink S_i are denoted by $E_{t_RTS}^{iS_i}$ and $E_{t_DATA}^{iS_i}$ respectively. If the rate of outgoing data from node i to j , including a neighbor or sink S_i is represented by r_{ij} , therefore the average consumed power of node i for sending data denoted by P_t^i would be as follows:

$$P_t^i = \sum_{j \in N_i^R + \{S_i\}} E_t^{ij} \cdot r_{ij} \tag{16}$$

Now we calculate the energy consumption related to waiting time during IL mode to receive the first CTS packet. When each node like i , wants to send a data packet, first it sends the RTS packet to its neighbors in radius R in flooding way. In this case the number of waking neighbors can be a number between 0 and N_i^R . Consider at the moment of sending RTS packet, the number of waking neighbors is k . In this situation, based on the protocol FRTS–RCTS in case 1, transmitter node i sends a random sequence of numbers 1 to N_i^R to all of its neighbors such that each of the random numbers is assigned to one of the neighbors. Since the transmitter node chooses the first CTS transmitter as the receiver of the data packet, we can consider the average of the receiving slot numbers as the receiving slot number of CTS packet on the whole working time of the network. Assume that X is a random variable used for showing the slot number in which the first CTS packet is received by node i . So based on different values of k the following cases would be occurred:

1. None of the neighbors are awake. In this case, the transmitter node does not receive any CTS packet during the CRS phase which is the time duration equivalent to $|N_i^R|$ time slots. So, the amount of waiting time would be equal to $|N_i^R|$ time slots.

- Among $|N_i^R|$ neighbors, only one neighbor is awake. In this case, it is clear that the random variable X is a uniform random variable in the range of $1-|N_i^R|$ and its average value will be calculated as follows:

$$E_1[X] = \sum_{n=1}^{|N_i^R|} n \cdot p(X = n) = \sum_{n=1}^{|N_i^R|} n \cdot \frac{1}{|N_i^R|} = \frac{|N_i^R| + 1}{2} \tag{17}$$

In (17), $E_1[X]$ has been used to show the expected value of random variable X when there is only 1 waking neighbor.

- Among $|N_i^R|$ neighbors, $k \geq 2$ neighbors are awake. In this situation, the random variable X takes one of the numbers between 1 and $|N_i^R| - k + 1$. In this case the probability that among k waking neighbors, CTS packet is received in the first slot of CRS phase can be computed as follows:

$$p(X = 1) = \frac{(|N_i^R| - 1) \cdot (|N_i^R| - 2) \dots (|N_i^R| - k + 1) \cdot k}{\binom{|N_i^R|}{k} \cdot k!}, \quad k \geq 2 \tag{18}$$

Also the probability that among k waking neighbors, CTS packet is received in the second slot of CRS phase can be computed as follows:

$$p(X = 2) = \frac{(|N_i^R| - 2) \cdot (|N_i^R| - 3) \dots (|N_i^R| - k) \cdot k}{\binom{|N_i^R|}{k} \cdot k!}, \quad k \geq 2 \tag{19}$$

And generally, the probability that among k waking neighbors, CTS packet is received in the n th slot of CRS phase can be written as follows:

$$\begin{aligned} p(X = n) &= \frac{(|N_i^R| - n) \cdot (|N_i^R| - n - 1) \dots (|N_i^R| - k + 2 - n) \cdot k}{\binom{|N_i^R|}{k} \cdot k!} \\ &= \frac{(|N_i^R| - n - m)}{\binom{|N_i^R|}{k} \cdot (k - 1)!}, \quad k \geq 2 \end{aligned} \tag{20}$$

Hence, the average of the receiving slot numbers will be calculated as follows:

$$\begin{aligned} E_k[X] &= \sum_{n=1}^{|N_i^R|-k+1} n \cdot p(X = n) \\ &= \sum_{n=1}^{|N_i^R|-k+1} n \cdot \frac{(|N_i^R| - n - m)}{\binom{|N_i^R|}{k} \cdot (k - 1)!}, \quad k \geq 2 \end{aligned} \tag{21}$$

If the time duration of each slot in CRS phase is denoted by t_s , the amount of waiting time for receiving the first CTS packet, in terms of all values k , represented by CRT as a function of variables N_i^R and k will be computed as follows:

$$CRT(N_i^R, k) = \begin{cases} |N_i^R| \cdot t_s & k = 0 \\ (E_k[X] - 1) \cdot t_s & k \geq 1 \end{cases} \tag{22}$$

In (22), function $CRT(N_i^R, k)$ computes waiting time for receiving the first CTS packet of node i containing N_i^R neighbors with k of them are awake. Let Ω_i^k be a k -member subset of N_i^R . The probability that among $|N_i^R|$ neighbors, only the neighbors belonging to Ω_i^k are awake at the sending moment of RTS denoted by $p(\Omega_i^k)$ is as follows:

$$p(\Omega_i^k) = \prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \tag{23}$$

If the average consumed power of each node in IL mode is P_{IL} , according to (23), the energy consumption of node i , in the waiting time of receiving the first CTS packet from its neighbors will be calculated as follows:

$$E_{w_CTS}^i = \sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \right] \cdot P_{IL} \tag{24}$$

Finally, the average consumed power of node i in the waiting time for receiving the first CTS packet from its neighbors in terms of data rate denoted by $P_{w_CTS}^i$ will be calculated as follows:

$$P_{w_CTS}^i = \left(\sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \right] \cdot P_{IL} \right) \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \tag{25}$$

In data receiving mode, if the receiver node not to be the first transmitter of CTS packet, thus won't receive any data packet in its corresponding time slot in DS Phase. The only energy consumption in this case is the energy consumed due to the hearing carrier signal in the channel at very short duration of time, which will be negligible. But, if a receiver node like k is the first transmitter of CTS packet, the energy consumption of node i is including energy consumed due to receiving a data packet from node k and consumed energy related to sending ACK packet. So we have:

$$E_{r_DATA}^{ki} = E_{r_DATA} + E_{t_ACK}^{ik} \tag{26}$$

If data receiving rate of node i from each node like k is r_{ki} , thus the average consumed power of node i in receiving data mode denoted by $P_{r_DATA}^i$ will be computed as follows:

$$P_{r_DATA}^i = \sum_{k:i \in N_i^R} E_{r_DATA}^{ki} \cdot r_{ki} \tag{27}$$

If a node like i receives a RTS packet, two possible cases would be occurred. Either node i will be the first transmitter of CTS packet or not. It is obvious that in both cases, it will send CTS packet to the transmitter. So, either in the first or second case, energy consumption denoted by E_{r_RTS,t_CTS}^{ik} would be computed as follows:

$$E_{r_RTS,t_CTS}^{ik} = E_{r_RTS} + E_{t_CTS}^{ik} \tag{28}$$

If sending data rate of every node like k , which node i is one of its neighbors, to all of its neighbors is denoted by $\sum_{j \in N_k^R} r_{kj}$ and node i works with duty cycle dc_i , therefore the share of receiving rate of RTS packets from node k to i denoted by $S_{r_CTS}^{ki}$ would be computed as follows:

$$S_{r_CTS}^{ki} = dc_i \cdot \sum_{j \in N_k^R} r_{kj} \tag{29}$$

So the average consumed power of node i in receiving RTS packets from node k and sending CTS packets to it which is denoted by P_{r_RTS,t_CTS}^{ik} would be calculated as follows:

$$P_{r_RTS,t_CTS}^{ik} = E_{r_RTS,t_CTS}^{ik} \cdot S_{r_CTS}^{ki} \tag{30}$$

Thus using (29) and (30), the average consumed power of node i in receiving all RTS packets from all of the neighbors that node i is their neighbor and sending all CTS packets to them denoted by P_{r_RTS,t_CTS}^i would be calculated as follows:

$$\begin{aligned} P_{r_RTS,t_CTS}^i &= \sum_{k:i \in N_k^R} P_{r_RTS,t_CTS}^{ik} = \sum_{k:i \in N_k^R} E_{r_RTS,t_CTS}^{ik} \cdot dc_i \cdot \sum_{j \in N_k^R} r_{kj} \\ &= dc_i \cdot \sum_{k:i \in N_k^R} E_{r_RTS,t_CTS}^{ik} \cdot \sum_{j \in N_k^R} r_{kj} \end{aligned} \tag{31}$$

In IL mode also, in which sensor node i listens to the radio channel, in order to receive any RTS packet, if it works with duty cycle of dc_i , its average consumed power will be as follows:

$$P_{IL}^i = P_{IL} \cdot dc_i \tag{32}$$

Finally, using Eqs. (16), (25), (27), (31) and (32) the average consumed power of each node like i denoted by \bar{P}_i^{cons} would be calculated as follows:

$$\begin{aligned} \bar{P}_i^{cons} &= \sum_{j \in N_i^R + \{S_i\}} E_i^{ij} \cdot r_{ij} + \left(\sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \right] \cdot P_{IL} \right) \\ &\cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} + dc_i \cdot \sum_{k:i \in N_k^R} E_{r_RTS,t_CTS}^{ik} \cdot \sum_{j \in N_k^R} r_{kj} + \sum_{k:i \in N_i^R} E_{r_DATA}^{ki} \cdot r_{ki} + P_{IL} \cdot dc_i \end{aligned} \tag{33}$$

2.4 EB-JRADCS Problem

In this section EB-JRADCS problem is constructed based on all of the above obtained constraints. If node i works in a time duration T_i and its initial battery energy is E_i , its energy constraint under a given flow rate $\mathbf{R}^i = \{r_{ij}\}$ and duty cycle dc_i will be written as follows:

$$\bar{P}_i^{cons} \cdot T_i(\mathbf{R}^i, dc_i) \leq E_i \tag{34}$$

Now, let us define the network lifetime as the minimum lifetime over all nodes, i.e.,

$$T_{net} = \min_{i \in V} T_i(\mathbf{R}^i, dc_i) \tag{35}$$

Our goal is to compute the flow data rates and duty cycle of all nodes such that the network lifetime is maximized. This maximization is carried out definitely by considering three aforementioned constraints including flow conservation constrains, FS constrains and energy constrains calculated in (1), (10), (11) and (33). Thus the EB-JRADCS maximization problem can be written as follows:

$$\begin{aligned} & \text{maximize} \left(\min_{i \in V} T_i(\mathbf{R}^i, dc_i) \right) \sum_{j \in N_i^R + \{S_i\}} r_{ij} - \sum_{k: i \in N_k^R} r_{ki} = g_i, \quad \forall i \in V \\ & \left[\sum_{j \in N_i^R + \{S_i\}} E_t^{ij} \cdot r_{ij} + \left(\sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \right] \cdot P_{IL} \right) \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \right. \\ & \left. + dc_i \cdot \sum_{k: i \in N_k^R} E_{r_RTS, t_CTS}^{ik} \cdot \sum_{j \in N_k^R} r_{kj} + \sum_{k: i \in N_i^R} E_{r_DATA}^{ki} \cdot r_{ki} + P_{IL} \cdot dc_i \right] \cdot T_i(\mathbf{R}^i, dc_i) \leq E_i \\ & r_{ij} = \left[\sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \\ & r_{iS_i} = \left[\prod_{k \in N_i^R} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \tag{36} \end{aligned}$$

Problem (36) is a max–min problem which can be rewritten as a maximization problem as follows:

$$\begin{aligned} & \text{maximize}(T) \sum_{j \in N_i^R + \{S_i\}} r_{ij} - \sum_{k: i \in N_k^R} r_{ki} = g_i, \quad \forall i \in V \\ & \left[\sum_{j \in N_i^R + \{S_i\}} E_t^{ij} \cdot r_{ij} + \left(\sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} (1 - dc_k) \right] \cdot P_{IL} \right) \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \right. \\ & \left. + dc_i \cdot \sum_{k: i \in N_k^R} E_{r_RTS, t_CTS}^{ik} \cdot \sum_{j \in N_k^R} r_{kj} + \sum_{k: i \in N_i^R} E_{r_DATA}^{ki} \cdot r_{ki} + P_{IL} \cdot dc_i \right] \cdot T \leq E_i \\ & r_{ij} = \left[\sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \\ & r_{iS_i} = \left[\prod_{k \in N_i^R} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \tag{37} \end{aligned}$$

In the next section we present the procedure of solving the EB-JRADCS problem.

3 Solving the EB-JRADCS Problem

It is clear that problem (37) is a highly nonlinear and non-convex problem generally. To solve this problem, first we try to conform it to one of the classic non-linear programming problems. To this end, first some primary preliminaries and definitions adopted from [13] are introduced in the beginning of this section.

Definition 1 A function $M(x)$ is monomial if

$$M(x) = a \cdot \prod_{k=1}^n x_k^{\alpha_k} \quad (38)$$

where a is a nonnegative constant coefficient and exponential constants α_k are real numbers.

Definition 2 A function $P(x)$ is posynomial if

$$P(x) = \sum_i M(x) \quad (39)$$

where the functions $M_i(x)$ are arbitrary monomial functions.

Definition 3 A function $\Phi(x)$ is called signomial if it is in the same form as a posynomial function except that multiplicative constant coefficients can be negative.

Definition 4: Signomial Geometric Programming (SGP) Problem A signomial geometric programming (SGP) problem is an optimization problem of the form:

$$\begin{cases} \min & \Phi_0(x) \\ \Phi_i(x) \leq 1, & \forall i = 1, \dots, p \\ \Phi_i(x) = 1, & \forall i = p + 1, p + 2, \dots, q \end{cases} \quad (40)$$

where function $\Phi_i(x)$ is signomial for all of the values $i \in [0, q]$. In special case, if $\Phi_i(x)$ is posynomial for $i \in [0, p]$ and monomial for $i \in [p + 1, q]$, the SGP problem is well known as standard geometric programming (GP) problem. In [15] it has been shown that a GP problem can be turned into an equivalent convex optimization problem and thus solved efficiently. But a SGP problem, in general, is an intrinsically intractable NP-hard problem and difficult to solve for global optimality. In [13, 16] two new methods have been introduced and presented in order to solve SGP problem. According to the above definitions, we can easily conform the EB-JRADCS problem into a SGP problem with replacing variable T by q^{-1} . Since at this problem objective function is posynomial and the constraints are as signomial functions. In the following we briefly illustrate the procedure of solving a SGP problem in the special case of posynomial objective function according to the method presented in [13]. The interested readers are referred to [13] for further details on solving the problem in general form and convergence proof. At this method, the signomial constraints should be converted into the standard constraints usable in a GP problem. Since the procedure of converting unequal signomial constraint is different from the equal one, at two the following parts the procedures are described briefly.

3.1 Unequal Signomial Constraints Conversion

Since in the standard GP problem, unequal constraints are posynomial, therefore unequal signomial constraints in SGP problem should be converted into posynomial terms. To this end, it is clear that a signomial function $\Phi_i(x)$ can be written as the difference of two posynomials $\Phi_i^+(x)$ and $\Phi_i^-(x)$ as follows:

$$\Phi_i(x) = \Phi_i^+(x) - \Phi_i^-(x) \tag{41}$$

In this case, unequal signomial $\Phi_i(x) \leq 1$ can be written as follows:

$$\frac{\Phi_i^+(x)}{1 + \Phi_i^-(x)} \leq 1 \tag{42}$$

Now to convert the left hand side of (42) into posynomial function the denominator of the fraction is approximated to a monomial function. For this purpose, consider a posynomial function as $P(x) = \sum_k M_k(x)$ in which functions $M_k(x)$ are monomial terms. The required monomial approximation can be calculated using the following arithmetic-geometric mean inequality [17].

$$g(x) \geq \hat{g}(x) = \prod_k \left(\frac{u_k(x)}{\alpha_k(x_0)} \right)^{\alpha_k(x_0)} \tag{43}$$

where parameters $\alpha_k(x_0)$ can be computed as follows:

$$\alpha_k(x_0) = \frac{u_k(x_0)}{g(x_0)} \quad \forall k \tag{44}$$

where x_0 is a positive and fixed point. It has been shown in [17] that the approximation of (43) is the best local monomial approximation of $P(x)$ near x_0 . Finally by applying the described monomial approximation to denominator of constraint (42), the constraint will be transformed into a posynomial function.

3.2 Equal Signomial Constraints Conversion

To convert equal signomial constraint as $\Phi_i(x) = 1$ into a standard constraint useable in GP problem, by using (41) two inequality constraints are constructed as follows:

$$\begin{cases} \frac{\Phi_i^+(x)}{1 + \Phi_i^-(x)} \leq 1 \\ \frac{\Phi_i^+(x)}{1 + \Phi_i^-(x)} \geq s_i^{-1}, \quad s_i \geq 1 \end{cases} \tag{45}$$

Or equivalently:

$$\begin{cases} \frac{\Phi_i^+(x)}{1 + \Phi_i^-(x)} \leq 1 \\ \frac{(1 + \Phi_i^-(x)) \cdot s_i^{-1}}{\Phi_i^+(x)} \leq 1, \quad s_i \geq 1 \end{cases} \tag{46}$$

In the inequalities (45) and (46), s_i is an auxiliary variable. As s_i approaches to 1, inequalities (46) will approach to equality and so $\Phi_i(x)$ will approach to 1. To accomplish this approach, all variables s_i is added to objective function as penalty term. Thus the new objective function can be rewritten as follows:

$$\Phi_0(x) + \sum_i w_i \cdot s_i \quad (47)$$

By using this policy when objective function is minimized s_i is forced to approach to 1 for all values i .

In (47), $w_i > 0$ are weighting coefficients with sufficiently large values. As it is seen, in the case of existence of any equality constraint, the converting process is more complicated than inequality one and according to [13], the rate of convergence will be slow.

Now, two unequal constraints in (46), based on what described in Sect. 3.1 are transformed into unequal posynomial constraints which are allowable expressions in GP problem. In fact this process is performed by means of converting the denominators of two unequal constraints into their equivalent monomials. Finally, after converting all unequal and equal signomial constraints, a standard GP problem is obtained which can be solved efficiently. Then in order to compute the global optimum solution, an iterative algorithm is used which has been introduced in [13]. It has been proven that the proposed iterative algorithm converges to the global optimum solution. Before presenting the algorithm, first we return to our problem and describe the procedure of converting it into the standard problem that will be solvable by this algorithm. In EB-JRADCS problem, flow conservation constraints and FS constraints are equal signomial constraints. So, according to Sect. 3.2 the constraints must be converted into equivalent approximate posynomial constraints. This procedure is almost easy for flow conservation constraints but because of being long terms in FS constraints the converting process would be quite complex. Also as it mentioned above, the rate of algorithm convergence will be so slow. In the following theorem, we show that these constraints can be relaxed to unequal form without any effect on the problem solution.

Theorem 1 *In EB-JRADCS problem, if equal constraints related to FS constraints are transformed into unequal constraints, the solution of the problem would not change.*

Proof To prove Theorem 1, first we state and prove the following lemma.

Lemma 1 *Suppose that x_i are non-negative variables and a_i are non-negative coefficients for all of the values $i = 1, 2, \dots, N$ such that $a_i \leq 1$ and two following expressions hold.*

$$x_i \geq a_i \cdot \sum_{n=1}^N x_n \quad \text{and} \quad \sum_{i=1}^N a_i = 1 \quad (48)$$

Then it can be concluded that for all of the values $i = 1, 2, \dots, N$, the following equalities hold:

$$x_i = a_i \cdot \sum_{n=1}^N x_n \quad (49)$$

To prove this lemma, we can use reductive ad absurdum. So let us consider the equality (49) is not true for only $i = j, j \in [1, N]$, Hence according to the assumption of the lemma we have:

$$x_i = a_i \cdot \sum_{n=1}^N x_n \quad \forall i = 1, 2, \dots, N \quad \text{and} \quad i \neq j \quad \text{and} \quad x_j > a_j \cdot \sum_{n=1}^N x_n \quad (50)$$

So we have:

$$x_1 = a_1 \cdot \sum_{n=1}^N x_n, \quad x_2 = a_2 \cdot \sum_{n=1}^N x_n, \dots, x_j > a_j \cdot \sum_{n=1}^N x_n, \dots, x_N = a_N \cdot \sum_{n=1}^N x_n \quad (51)$$

By adding all of the terms in the both sides of the relations in (51), we conclude:

$$\sum_{i=1}^N x_i > \sum_{i=1}^N a_i \cdot \sum_{n=1}^N x_n \Rightarrow 0 > 0 \quad (52)$$

But as it is seen the above resulted inequality leads to contradiction. Hence the assumption (50) is wrong and (49) will be true for all of the values $i = 1, 2, \dots, N..$

Now to prove Theorem 1, we define coefficients a_j as follows:

$$a_j = \begin{cases} \frac{1}{\sum_{WN_i^j \subseteq N_i^R} |WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) & j \in N_i^R \\ \prod_{k \in N_i^R} (1 - dc_k) & j = S_i \end{cases} \quad (53)$$

But according to (6) we have:

$$\sum_{j \in N_i^R + \{S_i\}} a_j = 1 \quad (54)$$

Now by applying the result of Lemma 1 and using (10) and (11), Theorem 1 is proved and so it concludes two bellow constraints:

$$r_{ij}^{-1} \cdot \left[\sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \leq 1 \quad (55)$$

$$r_{iS_i}^{-1} \cdot \left[\prod_{k \in N_i^R} (1 - dc_k) \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \leq 1 \quad (56)$$

Now, using (55) and (56), in the EB-JRADCS Problem, energy constraints and FS constraints are as unequal signomial constraints which must be converted into posynomials. But due to largeness of these constraints, this process will be complicated even in the

network with a few nodes. To solve this complication, we use a small variable changing in these constraints called off cycle (oc) as $oc = 1 - dc$. This only will add following auxiliary equal constraints to the problem.

$$dc_i + oc_i = 1 \quad \forall i \in V \tag{57}$$

By applying this variable changing, FS constraints and also energy constraints are converted into posynomial and thus don't need any extra converting. The only required conversion to solve the problem is converting equal flow conservation constraints and described auxiliary constraints in (57) into their approximate equivalences as has been illustrated in Sect. 3.2. In order to convert flow conservation constraints we can write:

$$\frac{\sum_{j \in N_i^R} r_{ij}}{g_i + \sum_{k: i \in N_k^R} r_{ki}} \leq 1, \quad \forall i \in V \Rightarrow \frac{\sum_{j \in N_i^R} r_{ij}}{\left(\frac{g_i}{\alpha_0}\right)^{\alpha_0} \cdot \prod_{k: i \in N_k^R} \left(\frac{r_{ki}}{\alpha_{ki}}\right)^{\alpha_{ki}}} \leq 1$$

$$\frac{\left(g_i + \sum_{k: i \in N_k^R} r_{ki}\right) \cdot s^{-1}}{\sum_{j \in N_i^R} r_{ij}}, \quad \forall i \in V \Rightarrow \frac{\left(g_i + \sum_{k: i \in N_k^R} r_{ki}\right) \cdot s^{-1}}{\prod_{j \in N_i^R} \left(\frac{r_{ij}}{\beta_{ij}}\right)} \leq 1 \tag{58}$$

Such that α_0 , α_{ki} and β_{ij} would be calculated as follows:

$$\alpha_0 = g_i / \left(g_i + \sum_{k: i \in N_k^R} r_{ki}^{(m)}\right), \quad \alpha_{ki} = r_{ki}^{(m)} / \left(g_i + \sum_{k: i \in N_k^R} r_{ki}^{(m)}\right),$$

$$\beta_{ij} = r_{ij}^{(m)} / \left(\sum_{j \in N_i^R} r_{ij}^{(m)}\right) \tag{59}$$

Similarly, to convert auxiliary constraints in (57), we have:

$$dc_i + oc_i \leq 1 \quad \forall i \in V \tag{60}$$

$$\frac{z_i^{-1}}{dc_i + oc_i} \leq 1 \quad \forall i \in V \Rightarrow \frac{z_i^{-1}}{\left(\frac{dc_i}{\gamma_i^1}\right)^{\gamma_i^1} \cdot \left(\frac{oc_i}{\gamma_i^2}\right)^{\gamma_i^2}} \leq 1 \tag{61}$$

Such that γ_i^1 and γ_i^2 would be calculated as:

$$\gamma_i^1 = dc_i^{(m)} / \left(dc_i^{(m)} + oc_i^{(m)}\right), \quad \gamma_i^2 = oc_i^{(m)} / \left(dc_i^{(m)} + oc_i^{(m)}\right) \tag{62}$$

In (59) and (62), $r_{ki}^{(m)} (\forall k), r_{ij}^{(m)} (\forall j), dc_i^{(m)}$ and $oc_i^{(m)}$ are given values that will be described later. Also s_i and z_i are auxiliary variables that will add to objective function as penalty terms.

Finally, EB-JRADCS problem for a given m will be rewritten as follows:

$$\begin{aligned}
 & \text{minimize } q + \sum_{i=1}^N w_i^{(m)} \cdot s_i + u_i^{(m)} \cdot z_i \\
 & \alpha_0 = g_i / \left(g_i + \sum_{k:i \in N_k^R} r_{ki}^{(m)} \right), \quad \alpha_{ki} = r_{ki}^{(m)} / \left(g_i + \sum_{k:i \in N_k^R} r_{ki}^{(m)} \right), \quad \beta_{ij} = r_{ij}^{(m)} / \left(\sum_{j \in N_i^R} r_{ij}^{(m)} \right) \\
 & \frac{\sum_{j \in N_i^R} r_{ij}}{\left(\frac{g_i}{\alpha_0} \right)^{z_0} \cdot \prod_{k:i \in N_k^R} \left(\frac{r_{ki}}{\beta_{ki}} \right)^{\alpha_{ki}}} \leq 1, \quad \frac{\left(g_i + \sum_{k:i \in N_k^R} r_{ki} \right) \cdot s^{-1}}{\prod_{j \in N_i^R} \left(\frac{r_{ij}}{\beta_{ij}} \right)^{\beta_{ij}}} \leq 1 \quad \forall i \in V \\
 & \frac{1}{E_i} \left[\sum_{j \in N_i^R + \{S_i\}} E_t^{ij} \cdot r_{ij} + \left(\sum_{k=0}^{|N_i^R|} CRT(N_i^R, k) \cdot \sum_{\Omega_i^k \subseteq N_i^R} \left[\prod_{k \in \Omega_i^k} dc_k \prod_{k \in N_i^R - \Omega_i^k} oc_k \right] \cdot P_{IL} \right) \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \right. \\
 & \quad \left. + dc_i \cdot \sum_{k:i \in N_k^R} E_{r_{RTS}, t_CTS}^{ik} \cdot \sum_{j \in N_k^R} r_{kj} + \sum_{k:i \in N_k^R} E_{r_DATA}^{ki} \cdot r_{ki} + P_{IL} \cdot dc_i \right] \cdot q^{-1} \leq 1 \quad \forall i \in V \\
 & r_{ij}^{-1} \cdot \left[\sum_{WN_i^j \subseteq N_i^R} \frac{1}{|WN_i^j|} \prod_{k \in WN_i^j} dc_k \prod_{k \in SN_i^j} oc_k \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \leq 1 \quad \forall i \in V \\
 & r_{iS_i}^{-1} \cdot \left[\prod_{k \in N_i^R} oc_k \right] \cdot \sum_{j \in N_i^R + \{S_i\}} r_{ij} \leq 1 \quad \forall i \in V \\
 & \gamma_i^1 = dc_i^{(m)} / \left(dc_i^{(m)} + oc_i^{(m)} \right), \quad \gamma_i^2 = oc_i^{(m)} / \left(dc_i^{(m)} + oc_i^{(m)} \right) \quad \forall i \in V \\
 & dc_i + oc_i \leq 1, \quad \frac{z_i^{-1}}{\left(\frac{dc_i}{\gamma_i^1} \right)^{\gamma_i^1} \cdot \left(\frac{oc_i}{\gamma_i^2} \right)^{\gamma_i^2}} \leq 1 \quad \forall i \in V \\
 & s_i \geq 1, \quad z_i \geq 1 \quad \forall i \in V
 \end{aligned} \tag{63}$$

Now we use iterative algorithm described in [13] and conform it to solve our problem. The steps of the adapted algorithm are given below:

Step 0 Set iteration counter $m = 0$. Choose initial feasible values for the variables $r_{ij}^{(0)}$, $dc_i^{(0)}$ and $oc_i^{(0)}$. For this purpose, one solution can be obtained by simply solving the simultaneous nonlinear Eqs. (1), (10) and (11). Also give initial weights $w_i^{(0)}$, $u_i^{(0)}$ and solution accuracy $\varepsilon > 0$.

Step 1 At the m th iteration, obtain all of the constraints in (63) with the given $r_{ij}^{(m-1)}$, $dc_i^{(m-1)}$ and $oc_i^{(m-1)}$.

Step 2 Solve the standard GP problem (63) to compute $r_{ij}^{(m)}$, $dc_i^{(m)}$ and $oc_i^{(m)}$ with weighting coefficients $w_i^{(m-1)}$, $u_i^{(m-1)}$.

Step 3 If the expressions $\left| r_{ij}^{(m)} - r_{ij}^{(m-1)} \right| \leq \varepsilon$, $\left| dc_i^{(m)} - dc_i^{(m-1)} \right| \leq \varepsilon$ and $\left| oc_i^{(m)} - oc_i^{(m-1)} \right| \leq \varepsilon$ are satisfied $\forall i \in V$ and $j \in N_i^R$ then stop the algorithm, else go to Step 4.

Table 1 Parameter settings [18]

Parameters	Value
e_{elec}	5×10^{-11} /bit
e_{amp}	1×10^{-11} J/bit/m $^\beta$
P_{IL}	1.25×10^{-5} J/s
β	3.5
E_i	21,600 J
RTS, CTS and ACK packet size	10 bytes
Data packet size	100 bytes
Data transmission rate	250 kbps
t_s	0.32 ms

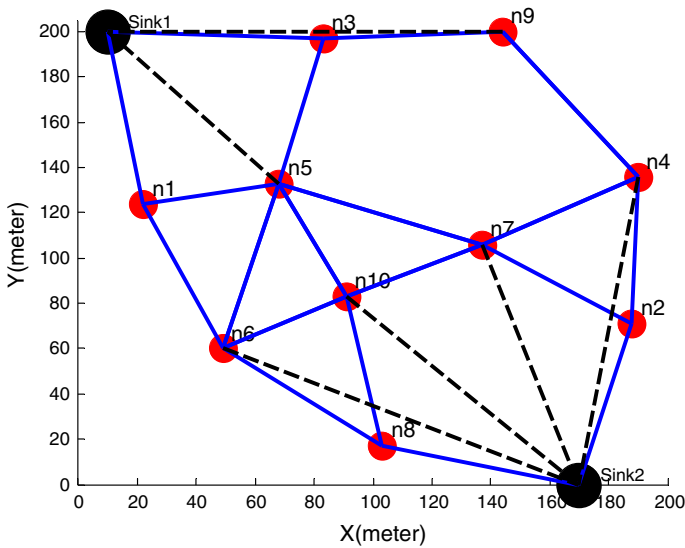


Fig. 3 Sensor network configuration with RTS flooding radius of 80 m

Step 4 Update the weighting coefficients $w_i^{(m)}, u_i^{(m)}$ with $w_i^{(m)} = W(w_i^{(m-1)})$, $u_i^{(m)} = U(u_i^{(m-1)})$; where W and U are monotonically increasing functions of $w_i^{(m-1)}$ and $u_i^{(m-1)}$ respectively. Set $m = m + 1$ and return to Step 1.

4 Simulation and Numerical Results

To simulate and evaluate this problem, we consider a sensor network including ten sensor nodes distributed in a square area of 200 m by 200 m. Also it is assumed that two sink nodes have been located at two separate positions of the area such that each sensor node is able to communicate at least with one of the sink nodes directly. We suppose that each sensor node has 3 V AA batteries with 2000 mAh capacity. All sensor nodes generate same and fixed data rate. The parameter settings for the simulation have been listed in

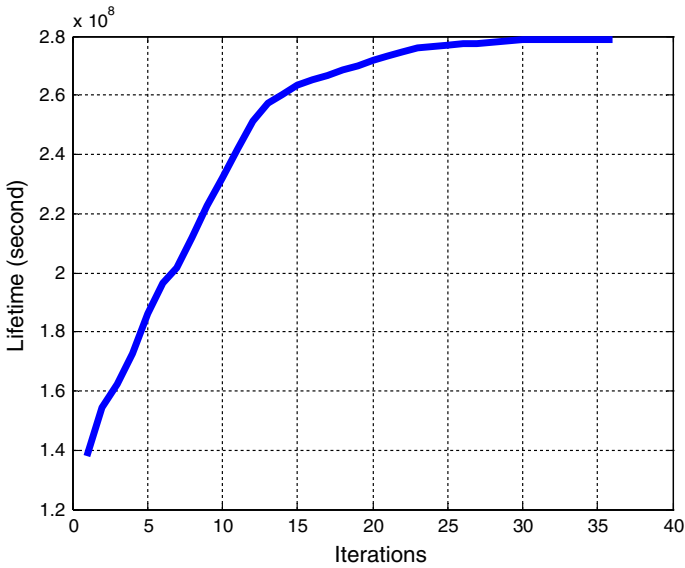


Fig. 4 Convergence of network lifetime to its maximum value

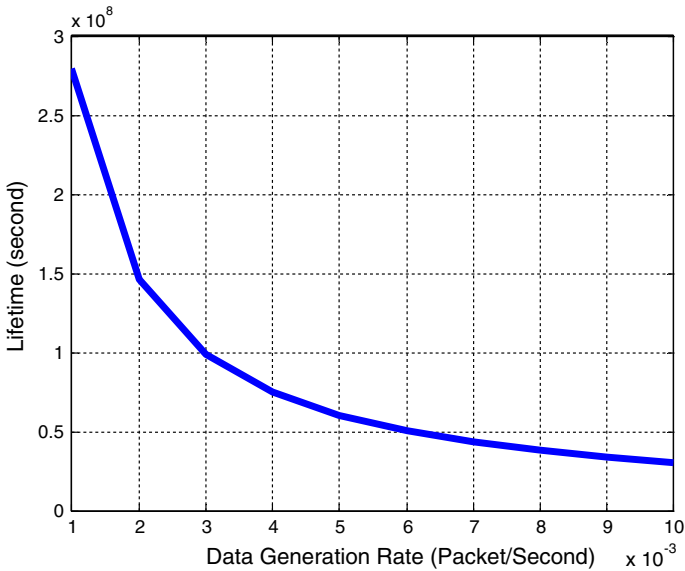


Fig. 5 Network lifetime versus data generation rate

Table 1. The parameters related to energy consumption have been selected similar to [18] based on Motes. To solve the problem (63) in each iteration, we use MATLAB based solver GPPLAB designed by Mutapcic et al. [19]. In Fig. 3, the configuration of the assumed network considering RTS flooding radius of 80 m has been shown. In this figure, each solid line shows a route between two sensor nodes or a route between one sensor node

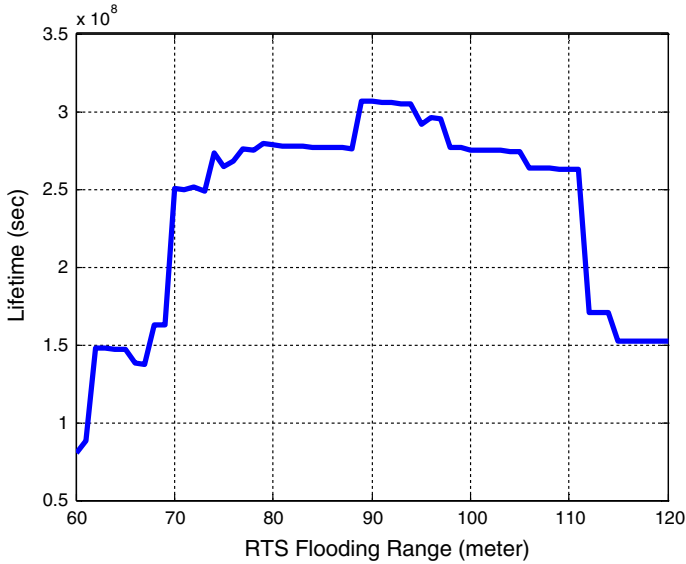


Fig. 6 Network lifetime versus RTS flooding radius

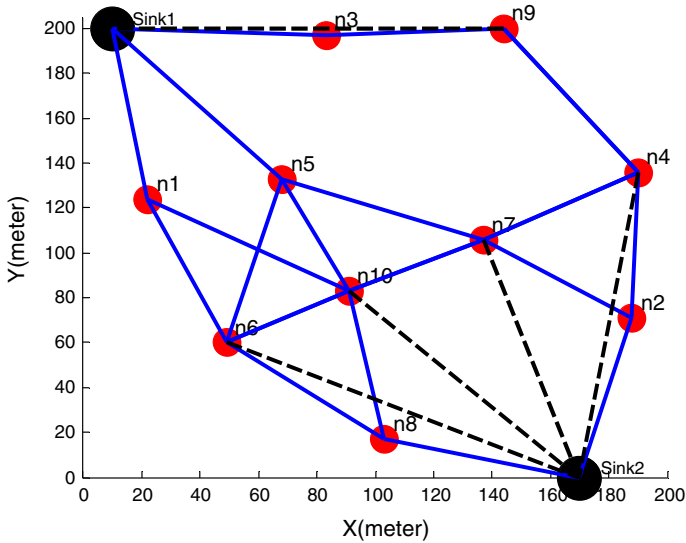


Fig. 7 Sensor network configuration with RTS flooding radius of 89 m

and the sink node which is in RTS flooding radius of the sensor node. Also each dash line shows a direct route between a sensor node and its nearest sink node which is not in its RTS flooding radius. It is clear that all of the obtained constraints in EB-JRADCS problem are quite dependent on RTS flooding radius. In the first experiment, we investigate that how the mentioned iterative algorithm converges to the optimum solution for a given RTS flooding radius. To this end, data generation rate is considered to be 0.001 packet/s and

RTS flooding radius is assumed to be 80 m. In Fig. 4 the optimum solution of EB-JRADCS problem means maximum network lifetime versus iteration number has been shown. As it is observed, the solution of the problem converges to its optimum level after 31 iterations. In the second experiment, we evaluate the effect of data generation rate on the network lifetime. In Fig. 5 the calculated maximum network lifetime versus various data generation rates from 0.001 to 0.1 packet/second and fixed RTS flooding radius of 80 m has been shown. As expected, by increasing data generation rate, network lifetime will decrease. Finally in the third experiment, we investigate the effect of changing RTS flooding radius on the maximum network lifetime. In Fig. 6, the network lifetime versus RTS flooding radius in the range of 60–120 m has been shown. As it is observed, first, the network lifetime increases by increasing RTS flooding radius until it reaches to its maximum level at radius 89 m and then by more increasing RTS flooding radius, network lifetime reduces. We can illustrate the reason of this result in the fact that by increasing RTS flooding radius, consumed energy for sending RTS packets will start to increase, but on the other hand less sensor nodes ought to be awake to receive RTS packets and so duty

Table 2 Flow rate of data packets on all routes, duty cycle of sensor nodes and their remaining energy

Node number	r_{i-j}	Flow rate ($\times 10^{-3}$)	Duty cycle	Remaining energy
1	$r_{1-sink1}$	1.678	0.4009	0
2	$r_{2-sink2}$	2.026	0.5532	0
3	$r_{3-sink1}$	1.960	0.8358	0
4	r_{4-2}	0.380	0	0
	r_{4-7}	0.288		
	r_{4-9}	0.149		
	$r_{4-sink2}$	0.183		
	$r_{5-sink1}$	1.185	0.0765	0
6	r_{6-1}	0.306	0.5541	0
	r_{6-5}	0.053		
	r_{6-8}	0.577		
	r_{6-10}	0.497		
	$r_{6-sink2}$	0.120		
7	r_{7-2}	0.645	0.4483	0
	r_{7-4}	0		
	r_{7-5}	0.070		
	r_{7-10}	0.708		
	$r_{7-sink2}$	0.289		
8	$r_{8-sink2}$	2.269	0.6578	0
9	r_{9-3}	0.962	0.2557	0
	r_{9-4}	0		
	$r_{9-sink1}$	0.188		
10	r_{10-1}	0.372	0.5912	0
	r_{10-5}	0.063		
	r_{10-6}	0.551		
	r_{10-7}	0.424		
	r_{10-8}	0.691		
	$r_{10-sink2}$	0.103		

cycle of them will reduce. Hence this reduction overcomes to increasing energy consumed due to increasing RTS flooding radius until a threshold level. After the threshold level this procedure will be occurred conversely. It means when RTS flooding radius increases more than the threshold level, the energy consumed due to this increasing will dominate to energy consumed due to duty cycle of nodes. In Fig. 7, the network configuration in the case of RTS flooding radius of 89 m has been shown.

In Table 2, the obtained flow rate of data packets, duty cycle of sensor nodes and remaining energy at the ending of network lifetime in the case of RTS flooding radius of 89 m and data generation rate of 0.001 packet/s has been shown. As it is seen, under the achieved flow rates and duty cycles, the amount of remaining energy in all of the sensor nodes has been equal to zero. It means that consumed energy has been balanced in all of the sensor nodes. In other words all of the nodes in the network run out of energy exactly at the same time equal to maximum network lifetime.

5 Conclusion

In this paper, the issue of lifetime maximization in WSNs based on Energy Balance, employing Joint routing and asynchronous duty cycle scheduling (EB-JRADCS) techniques was introduced. This problem was modeled based on a new asynchronous MAC protocol named FRTS-RCTS. This protocol was the modified version of traditional CSMA/CA protocol which supported the duty cycle scheduling mechanism. With the aid of the proposed protocol new constraints named as FS constraints, were found which jointed network lifetime maximization parameters including flow rate of information on any route and duty cycle of nodes. Also the precise computation of energy consumption based on the FRTS-RCTS protocol was accomplished. With the small change of a parameter in the formulated EB-JRADCS problem, it was converted into the SGP problem and due to the complexity of solving the resulted problem, first it was transformed into a simpler problem by relaxing FS constraints from equal to unequal form. Then the simplified problem was solved utilizing a specific convexification method and the global optimum solution of the network lifetime was evaluated under various scenarios. The simulation results verified that under achieved flow rates and duty cycles, the amount of remaining energy in all of the sensor nodes had been equal to zero. In other words, the consumed energy had been balanced in all of the sensor nodes. The achieved optimum solution can be used as a benchmark for evaluating and comparing distributed and heuristic methods that aim to increase the network lifetime. In the future work we are going to find a heuristic distributed algorithm that will use only some local information to adjust duty cycle of nodes which in turn will lead to calculation of data flow rates in all of the routes based on found FS constraints in this paper.

References

1. Chang, J.-H., & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4), 609–619.
2. Kacimi, R., Dhaou, R., & Beylot, A.-L. (2013). Load balancing techniques for lifetime maximizing in wireless sensor networks. *Ad Hoc Networks*, 11(8), 2172–2186.
3. Ok, C.-S., Lee, S., Mitra, P., & Kumara, S. (2009). Distributed energy balanced routing for wireless sensor networks. *Computers & Industrial Engineering*, 57(1), 125–135.

4. Yardibi, T., & Karasan, E. (2010). A distributed activity scheduling algorithm for wireless sensor networks with partial coverage. *Wireless Networks*, 16(1), 213–225.
5. Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537–568.
6. Madan, R., & Lall, S. (2006). Distributed algorithms for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(8), 2185–2193.
7. Anastasi, G., Conti, M., & Di Francesco, M. (2009). Extending the lifetime of wireless sensor networks through adaptive sleep. *IEEE Transactions on Industrial Informatics*, 5(3), 351–365.
8. Wei, Y., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3), 493–506.
9. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on embedded networked sensor systems*, Baltimore, USA (pp. 95–107).
10. Tae, P., Kyung-Joon, P., & Lee, M. J. (2009). Design and analysis of asynchronous wakeup for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 8(11), 5530–5541.
11. Kartal Cetin, B., Prasad, N. R., & Prasad, R. (2013). Maximum lifetime routing problem in duty-cycling sensor networks. *Wireless Personal Communications*, 72(1), 101–119.
12. Liu, F., Tsui, C.-Y., & Zhang, Y. J. (2010). Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks. *IEEE Transactions on Wireless Communications*, 9(7), 2258–2267.
13. Xu, G. (2014). Global optimization of signomial geometric programming problems. *European Journal of Operational Research*, 233(3), 500–510.
14. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), 660–670.
15. Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.
16. Hou, X.-P., Shen, P.-P., & Chen, Y.-Q. (2014). A Global optimization algorithm for signomial geometric programming problem. *Abstract and Applied Analysis*,. doi:10.1155/2014/163263.
17. Boyd, S., Kim, S.-J., Vandenberghe, L., & Hassibi, A. (2007). A tutorial on geometric programming. *Optimization and Engineering*, 8(1), 67–127.
18. Xiaoguang, Z., & Zheng Da, W. (2010). Energy balanced routing strategy in wireless sensor networks. In *IEEE/IFIP 8th international conference on embedded and ubiquitous computing (EUC)* (pp. 436–443).
19. Mutapcic, A., Koh, K., Kim, S., Vandenberghe, L., & Boyd, S. (2006). *A matlab toolbox for geometric programming*. <http://stanford.edu/~boyd/ggplab/>.



Mohsen Kariman-Khorasani was born in Mashhad, Iran, in 1980. He received the B.Sc. degree in electrical engineering from Ferdowsi University, Mashhad, Iran, in 2003 and the M.Sc. degree in Telecommunications engineering from Ferdowsi University, Mashhad, Iran, in 2006. He joined the Electrical Engineering Department of Islamic Azad University, Gonabad, Iran In 2007. Now he is a Ph.D. student at Science and Research Branch, Islamic Azad University, Tehran (SRBIAUT), Iran. His research interests include designing of MAC layer protocols, efficient routing algorithms, lifetime optimization in wireless sensor networks and cloud computing.



Mohammad Ali Pourmina received his B.Sc. and M.Sc. degrees in electrical engineering (Telecommunication) from Islamic Azad University, Central branch, Tehran, Iran, in 1989 and 1991 respectively. He received his Ph.D. degree at Science and research Branch, Islamic Azad University, Tehran, Iran, in 1996. He joined Science and research Branch, Islamic Azad University, Tehran, Iran in 1996. He has been a member of Iran Telecommunication research center since 1992. He has performed research in the areas of packet radio networks and digital signal processing systems since 1991. His current research interests include spread-spectrum systems, cellular mobile communications, indoor wireless communications, DSP processors and wireless multimedia networks.



Ahmad Salahi was born in Tehran, Iran, in 1947. He received his B.Sc. degree in electrical engineering from Tehran University, Iran, his M.Sc. degree from Kansas University USA in 1974 and his Ph.D. degree from Purdue University West Lafayette Indiana, USA in 1979, all in electrical engineering. At present, he is a senior project manager in Network Security Department in Iranian Research Institute for ICT (ex. ITRC). His research interests are network security, switching and routing.