TÜBİTAK

Turk J Elec Eng & Comp Sci, Vol.20, No.6, 2012, © TÜBİTAK
doi:10.3906/elk-1012-5

Localized power-aware routing with an energy-efficient pipelined wakeup schedule for wireless sensor networks

Taner ÇEVİK[1,*], **Abdul Halim ZAİM**[2], **Derya YILTAŞ**[3]

[1] *Department of Computer Engineering, Fatih University, Büyükçekmece, İstanbul-TURKEY*
e-mail: tcevik@fatih.edu.tr

[2] *Department of Computer Engineering, İstanbul Commerce University, Avcılar, İstanbul-TURKEY*
e-mail: azaim@iticu.edu.tr

[3] *Department of Computer Engineering, İstanbul University, Avcılar, İstanbul-TURKEY*
e-mail: dyiltas@istanbul.edu.tr

Received: 02.12.2010

Abstract

Depending on the evolutions in electronics, wireless sensor networks (WSNs) have been very popular in many areas of human life such as health, industry, and military. This popularity has drawn the attention of many researchers toward WSNs. WSNs are especially favorable in conditions in which it is physically difficult and dangerous for human beings to gather information. Therefore, the lifetime of those networks must be as long as possible, since it would also be infeasible to replace the energy-depleted sensors with new ones, as they may be deployed in such geographical areas that are difficult and dangerous for human beings to enter. In this paper, we present a localized energy-aware routing method, LEERA-MS, and an alternative LEERA-MS-TH method, used in cooperation with an energy-efficient sleep-wakeup schedule, which is included with the pipelining mechanism that we previously proposed. Employing multiple sinks improves performance by providing a fair distribution of the load. Simulation results show that this routing method, applied on a multiple sink topology and when employed together with the pipelined sleep-wakeup schedule, provides 40% longer lifetime for WSNs.

Key Words: *Energy efficiency, power aware, routing wireless sensor networks*

1. Introduction

Developments in electronics and hardware technology have caused the emergence of new devices called sensors. These tiny devices mainly consist of a sensing mechanism, a processor, and a communication mechanism. Physical data are gathered by the sensing mechanism and transmitted to the data collection center, called a sink. These sensors and their communication protocols form wireless sensor networks (WSNs). WSNs generate

*Corresponding author: Department of Computer Engineering, Fatih University, Büyükçekmece, İstanbul-TURKEY

a class of ad hoc networks. However, they differ from traditional ad hoc networks in many ways. First of all, the nodes in WSNs are deployed in a more intensive manner than the nodes in traditional ad hoc networks. Moreover, nodes in most of the traditional ad hoc networks communicate in a point-to-point manner [1]. However, since sensor nodes are small energy constraint devices, their power sources and radio transmission ranges are limited. The radio performances of sensor nodes in both indoor and outdoor environments have been investigated by Matthew et al. [2]. According to their experiments, if both communicating nodes reside on the same plane, the coverage area is approximately 13.72 m. If one is located at a higher position than the other, then the coverage area widens to 30.48 m. Otherwise, if they are both raised up from the floor to a higher position, the coverage distance widens to 45.72 m. Therefore, data collected by the sensor nodes cannot be directly transmitted to the sink. They have to be relayed by multiple other interrelaying nodes in order to reach to the sink. Thus, a multihop communication manner is used, which is solely based on broadcasting. Furthermore, in some application areas, the mobility of WSNs is higher. If the differences mentioned above are taken into account, it can be easily noticed that not every method and protocol used for traditional ad hoc networks can be applied to all WSN applications [1].

Many studies have been done with the aim of developing efficient protocols and methods for different types of WSN applications. Actually, the major problem with WSNs is the quick energy depletion of the sensor nodes. To date, researchers have proposed different methods that aimed to prevent redundant energy consumption in a way such as in [3]. As described in [4], those techniques are roughly grouped into 3 categories: duty-cycling methods, data-driven approaches, and mobility.

The main idea of duty-cycling is to prevent all of the network nodes from staying awake continuously. This can be achieved in 2 ways. These solutions are not alternatives to each other, but rather they are complementary. The first method is topology control. In this method, all of the nodes do not need to remain continuously active. Instead, only a portion is handled awake in order to ensure the connectivity in the network. Moreover, the sensor nodes spend most of their time sensing the environment. During that time, the sensor node radios are redundantly turned on. It was mentioned in [5] that nodes consume nearly the same amount of energy during the transmit, receive, and idle states. Therefore, it is unnecessary to leave the sensor node radios in the 'on' state during idle periods. The second method in duty-cycling defines schedules for sensor nodes that represent their sleep and wakeup cycles.

Data-driven approaches can be roughly divided into 2 classes as data reduction and energy-efficient data acquisition techniques [4]. All of the methods grouped in these classes aim to reduce the amount of data that will be delivered to the sink.

For the reasons mentioned above, routing techniques used in traditional ad hoc networks are not suitable for WSNs. Hence, energy-efficient routing methods must be developed for WSNs.

2. Related work

Conventional routing methods used in traditional wired or wireless networks are not suitable for WSNs. Since wired networks and other ad hoc wireless networks do not have energy problems, most of the routing algorithms at work in those networks aim to find the best path, such as the shortest path or the one with the maximum bandwidth for transmission. It is obvious that latency is very important for some applications, such as in military or health. However, not all applications are so sensitive to delays. For such applications, routing algorithms should be chosen so as to construct paths that help to prolong the network lifetime. If the shortest path algorithm is chosen as the routing algorithm, packets emerging from a node always follow the same path.

Nevertheless, following the same path always includes the same nodes and those nodes deplete energy more quickly than the others. Thus, the main purpose of choosing the routing method in WSNs must be to provide a load balance. In other words, the routing algorithm should be able to find a path that passes via the nodes that have more energy than the others. This technique might extend the time it takes for the packet to reach the sink, but it also prolongs the lifetime of the network.

As described in [1] comprehensively, a node can choose one of the specified paths. One of the paths is the maximum power available route. The total available power is calculated by adding all available powers of nodes along the routes. Another alternative route chooses the path that consumes the minimum amount of energy while transmitting packets from the source to the sink. Moreover, the path including the minimum number of hops can be chosen. Finally, the path containing the node with the minimum amount of remaining power between all of the nodes among the other paths can be followed.

Flooding is a very old technique that can be used in WSNs. However, it is not preferred since it is not power efficient. A node with a packet to transmit broadcasts the packet to all of its neighbors. Receiver neighbors broadcast that packet to all of their neighbors again and the process iterates until the packet reaches the sink.

Gossiping [6] is another routing technique in which a node randomly selects a neighbor as the next hop. Since criteria like energy or delay are not considered when choosing the next hop, it is not a favorable routing algorithm for WSNs.

Greedy perimeter stateless routing [7] uses geographical positions of nodes for the next hop decision. Each node in the network is assumed to know its immediate neighbors' geographical positions. Before sending the packet, the transmitting node calculates the distance between all of its neighbors and the sink. It forwards the packet to the neighbor that is closest to the sink.

Another energy-aware routing algorithm is the localized energy-aware restricted neighborhood for ad hoc networks (LEARN) [8]. In this method, the sender node chooses a neighbor that has the largest distance from it in a particular area. This idea may not be valid every time. Traditional greedy forwarding [7] is applied to some situations in which this theory does not work.

In sensor protocols for information via negotiation (SPIN) [9], a node first broadcasts an advertisement for a packet before sending it. This advertisement describes the data in the packet. Nodes dealing with the data packet send a request back to the sender. After that, the sender node broadcasts the data packet to the demanding nodes.

Low-energy adaptive clustering hierarchy (LEACH) [10] is an energy-efficient clustering-based protocol. In LEACH, the nodes in the network maintain their missions in 2 states, the setup and steady phases. In the setup phase, the heads of the clusters are selected randomly. After head selection, the network goes into the steady phase, in which transmissions take place. Every node determines which cluster head it will belong to. After that, cluster heads define a schedule and announce this schedule to the nodes in its cluster. Nodes go into the sleep state, except during the transmission time selected for them by the cluster head. During the transmitting time, if they have packets to send, they transmit their packets to the cluster heads. After the cluster heads get all of the packets from the nodes in their clusters, they aggregate and compress them. Finally, they start to send this aggregated data to the sink.

Power-efficient gathering in sensor information systems (PEGASIS) [11] is constructed over a chain structure. This protocol is mainly based on LEACH [10]. Every node in the network is assumed to have global knowledge of the network. Depending on this assumption, it is easy to construct the chain using the greedy algorithm. However, PEGASIS does not show sufficient efficiency for sensor networks because the global network information to be carried to all of the nodes in the networks causes too much overhead.

A novel location-based algorithm for energy balance in WSNs is proposed in [12]. In this method, both the closeness of the hops to the sink and their residual energies are considered when choosing the next hop. However, the route construction method is not clearly defined in this study. Moreover, how the residual energy updates are done is also not clearly identified. Both the route construction messages and the residual energy updates bring too much overhead to the network.

In this paper, the energy gain provided by the pipelined sleep-wakeup schedule that we have proposed before is developed further by combining it with a localized power-efficient, load-balancing routing method. In most of the methods that consider the neighborhood energy level locally, the sender node requires the knowledge of the remaining energy levels of the neighbors in the coverage. This knowledge is provided by periodically broadcasting the energy levels between the nodes or by making estimations. Transmitting the energy levels periodically or on demand brings too much overhead to the network and causes redundant energy consumption. However, with the method that we propose, there is no redundant overhead of transmitting energy level updates. Furthermore, in order to ensure a fair load distribution on the nodes, it is necessary for the consequent transmissions to follow a distinct path as much as possible. This method prolongs the network lifetime by almost 40% when including the wakeup schedule that we proposed previously [13,14].

3. Localized power-aware routing with an energy-efficient pipelined wakeup schedule (LEERA-MS)

3.1. Sleep-wakeup schedule

In many types of WSNs, the nodes spend most of their time sensing the environment. As soon as a sensor detects an event, it leaves the idle state and starts to transmit related data toward the sink. In our approach, the communication channel is partitioned into 2 separate channels. The first is the control channel, used by nodes for transmitting their signaling messages on, and the other is the data channel, which is used only for data transmission. The reason for separating the whole bandwidth into 2 distinct channels is to prevent contention between the data and control information. In order to utilize 2 distinct channels simultaneously, every node in the network has 2 radios assigned statically to one of the channels described above. Since the control channel is used only for informing neighbor nodes about forthcoming transmissions and the candidate receiving node, which should be awake at that time, that channel can also be called the wakeup channel, as it is presented in Figure 1.

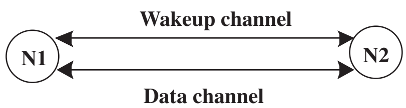

Figure 1. Signaling and data channels.

An 802.11 channel access mechanism is employed for the coordinating access of the nodes to the control channel. A contention mechanism is preferred here rather than time division multiple access, because all of the sensors do not always have data to transmit. Thus, it would be a waste of bandwidth to assign slots to the inactive nodes redundantly. Figure 2 represents the situation where 3 nodes communicate in a flat manner.

$Node_a$ wants to send data to $Node_b$, but it must wait for the signaling channel wakeup period. When the wakeup time comes, it sends a request to send (RTS) message to its neighborhood. In this message, there is the identification (ID) of the receiver of the forthcoming data transmission, which is $Node_b$, and also the amount of transmission time the data will take. At this time, all of the nodes within communication range of $Node_a$ take this message via their radios assigned for the control channel. By the time $Node_b$ gets the RTS message, it replies with a clear to send (CTS) message and turns its data radio on. By the reception of those RTS and CTS messages, the neighbors of both the sender and receiver nodes are informed about the forthcoming transmission. Thus, they will be informed about times at which they should not attempt to access the communication channel and to not stay awake redundantly. It is obvious that, until the end of the ongoing transmission, neighboring nodes in the communication range will not be able to transmit or receive anything. Otherwise, collision will occur. Therefore, it is senseless to leave communication radios on unnecessarily during that time. Hence, all of the other neighboring nodes turn their radios off, except for $Node_b$. $Node_a$ starts the transmitting process. After $Node_b$ receives all of the data, it cannot immediately transmit it to the next hop, $Node_c$, over the data channel, because the data radio of the next hop, denoted by $Node_c$, is turned off. Thus, it has to wait until the next wakeup time to start the process again.

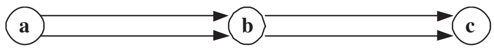

Figure 2. Simple scenario.

State transitions of all 3 nodes are shown in Figure 3.

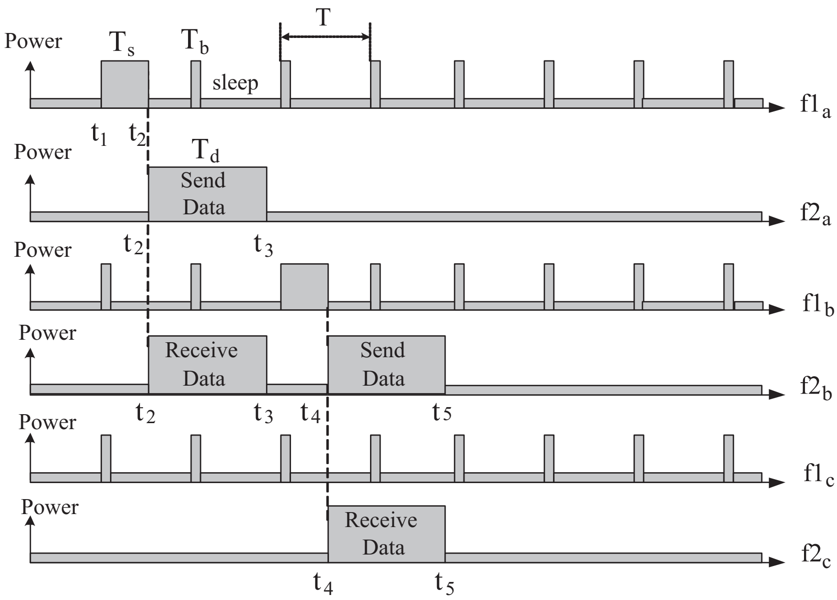

Figure 3. Send-receive operations for sparse topology and energy management (STEM).

As mentioned above, in order for $Node_b$ to start the transmission process immediately, the next hop, denoted by $Node_c$, must be aware of the forthcoming transmission. Moreover, nodes other than $Node_c$ should not turn their data radios and control channel radios on unnecessarily if there is a forthcoming transmission in their neighborhood not destined for them. With the proposed sparse topology and energy management (STEM)

[15] method, $Node_b$ has to wait until the next wakeup time. However, if $Node_c$ can somehow be informed about when $Node_b$ will start to transmit, $Node_c$ can turn its data radio on without waiting for the next wakeup time. By adding the pipelining approach to the STEM method, that unnecessary waiting time until the next wakeup time is prevented. When $Node_b$ sends the CTS response message back to $Node_a$, it immediately identifies the next receiver of the data packet and indicates the ID of the next hop in the response message. Since the communication medium is a broadcasting environment, all of the neighbors of $Node_b$ will also get this message and they will be informed about the next hop of the data packet after $Node_b$, which is $Node_c$, and about the amount of time it will take for the transmission. This allows $Node_c$ to arrange its wakeup time according the time stamp indicated in the CTS message. Other neighbors of $Node_b$ will not wake up unnecessarily at the next wakeup time if the forthcoming transmission does not end at that time, because they are not the owner of the channel at that time and cannot transmit or receive anything. Otherwise, they will collide with the ongoing transmission between $Node_b$ and $Node_c$.

By adding pipelining, $Node_b$ will not have to wait for $Node_c$ to wake up, as is shown in Figure 4, where all delays, such as propagation and processing delays, are ignored.

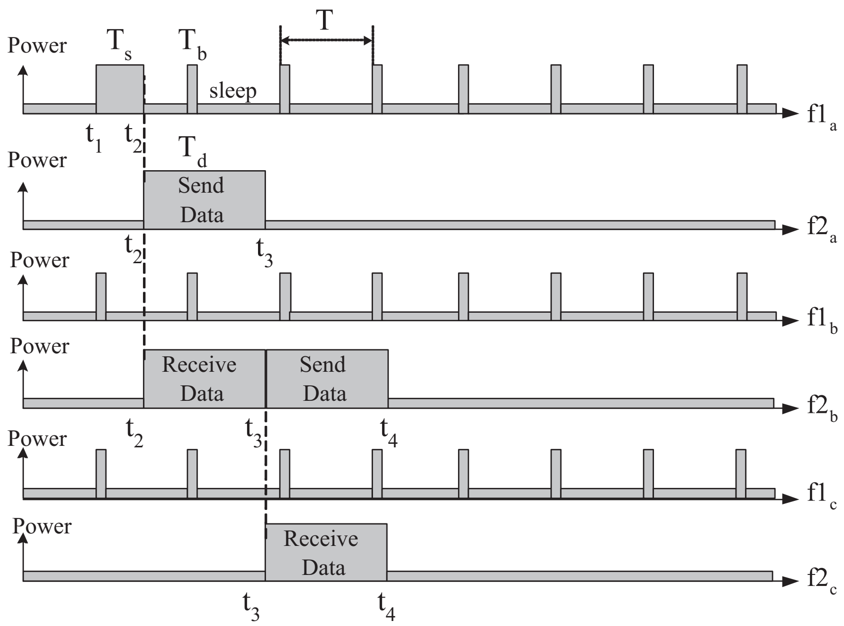

Figure 4. Send-receive operations for STEM with pipelining.

It is waste of energy for sensor radios to redundantly stay awake during idle times. With STEM [15], nodes periodically turn their signaling radios 'on.' At other times, both the signaling radio and the data radios are turned off. However, waiting for the next beacon period for transmitting data causes latency. As proposed in [13,14], by including pipelining in STEM, a significant amount of improvement in latency is achieved, as is clear in Figure 5.

In Figure 5, the transmission delay of the packet is observed for both STEM and pipelined STEM, employing 4 different routing methods. The first is closest to the sink (CTTS); it is also called the shortest path. In this routing method, the neighbor that is the closest to the sink is chosen as the next hop. CAN in Figure 5 denotes the node with the closest angled node to the sink. With the farthest node (FN), the neighbor that is the farthest from the sending node is chosen. Finally, NN represents the nearest node method, in which the neighbor that is closest to the sending node is chosen as the next hop.

3.2. Load balance routing (LEERA)

The main challenge to be considered with WSNs is designing and developing energy-efficient communication mechanisms. Designing energy-efficient routing methods is one of the approaches employed with the aim of prolonging the lifetime of the networks. Traditional routing methods used in wired networks, like the shortest path algorithm, cannot be employed here because by using the shortest path algorithm, all of the packets emerging from a node will follow the same path on the way to the sink, as shown in Figure 6.

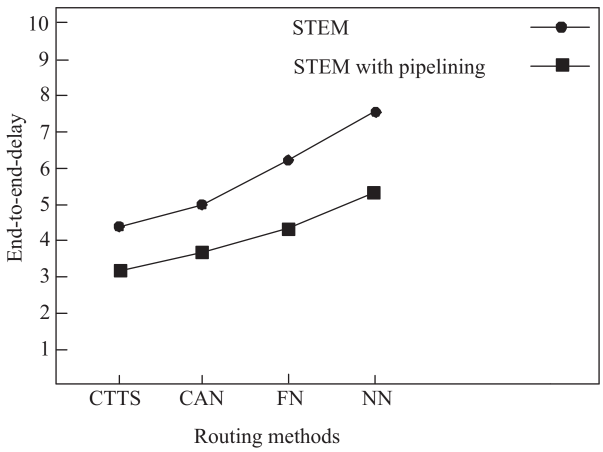

Figure 5. End-to-end-delay with STEM and pipelined STEM.

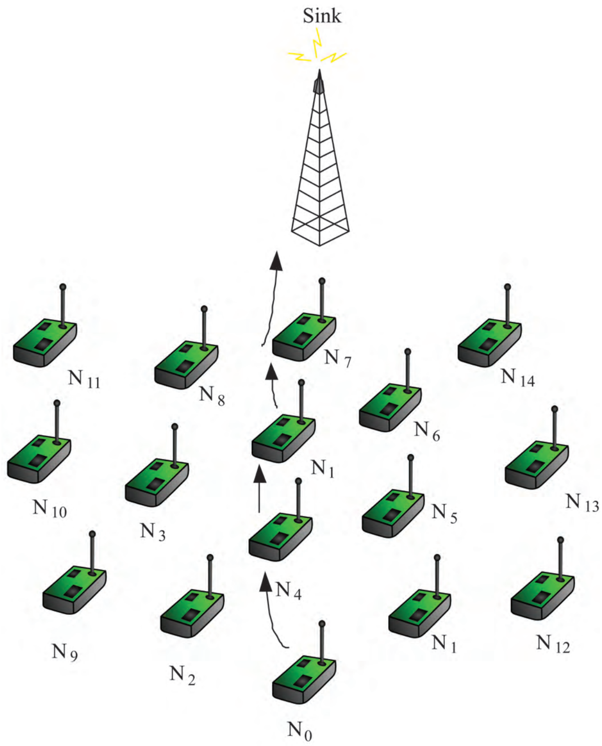

Figure 6. Sample topology.

Obviously, packets emerging from N_0 will follow the path $N_4 \diamondsuit N_1 \ \diamondsuit N_7$ if the shortest path algorithm is used as the routing method. Thus, forwarding nodes on the path will be depleted of energy quickly. Instead, a method should somehow be developed in order to not always use the same nodes while relaying the packets to the sink. The packet-relaying load will thereby be distributed over different nodes in an effort to prevent the nodes from being depleted of energy quickly.

Several methods have been proposed for energy-efficient routing in WSNs. Two main approaches have been applied. First is the global knowledge about the energy levels of the nodes in the topology. In order to provide that global knowledge to every node in the network, many messages should be carried inside the network. Those informational messages can be sent periodically or when an energy level change occurs. Link state or distance vector types of algorithms can be applied in this state. However, while transmitting this information between the nodes all over the network, there will be incredible energy waste, which is the main challenge and problem to be considered and solved in WSNs. The second approach is making the routing decisions according to local information about the nodes in their communication ranges. Giving routing decisions according to local knowledge can be assured in 2 ways. One is the way in which the nodes periodically probe their residual energy levels, which is also an energy-consuming procedure. The other alternative method is the one that we propose here, recalculating the residual energy level of the sender and the receiver nodes by the information overheard in the RTS and CTS messages. By employing the RTS/CTS mechanism, every node within communication range of a communicating pair of nodes is informed about the details of the forthcoming transmission. Since the amount of data is identified in the RTS and CTS messages, all of the neighbors of both the sender and the receiver can calculate how much energy will be consumed by both the sender and the receiver according to the formulas given below:

$$E_{Transmit} = T_d * P_{Transmit}, \tag{1}$$

$$E_{Receive} = T_d * P_{Receive}. \tag{2}$$

Here, $E_{Transmit}$ and $E_{Receive}$ describe the energy consumed by the receiving and transmitting nodes, respectively. T_d denotes the data transmission period, and $P_{Transmit}$ and $P_{Receive}$ describe the power levels required by the transmitter and the receiver. Nodes overhearing the RTS and CTS messages make the calculations mentioned above and update the relevant field, denoted by *EnergyConsumed* in their routing tables. In Figure 7, N_0 wants to transmit data toward the sink. It waits for the next beacon period in the signaling channel. When the time comes, it checks its routing table and chooses the neighbor that has the greatest energy level. Actually, in our approach, the nodes do not keep the residual energy levels; instead, when a transmission occurs, the amount of energy consumed is added to the energy fields of the records belong to the transmitting and receiving nodes. Therefore, the next hop for the data packet is identified by searching for the node in the table that has consumed the least amount of energy up to that time. If there is more than one such node, then a second criterion is employed in order to search for the next node. All of the nodes in the network are assumed to know their geographical positions as well as the geographical position of the sink and the nodes in their communication range. It is possible to provide that information by equipping the nodes with a GPS device and transmitting the geographical positions of the nodes between them at the setup state. The distance between the node and the sink is calculated according to the Euclidean formula.

At the beginning, since all of the neighbors of N_0 have the same amount of residual energy, the first packet emerging from N_0 will follow the shortest path, i.e. $N_4 \diamondsuit N_1 \ \diamondsuit N_7$. For the second packet, the same path is not used again. Since N_4 is employed in the forwarding process of the first packet, other neighbors of N_0 should be used this time. The one that is closer to the sink is chosen as the next hop. Let us assume that N_1 is closer and is chosen as the next hop. The next hop for the second packet is chosen through nodes N_4, N_5, and N_{13}. Since N_4 is employed in the forwarding process of the first packet, it should be exempt this time. Hence, there are 2 alternatives, N_5 and N_{13}. N_5 is closer to the sink than N_{13} and is chosen as the next hop. The next hop for the second packet is chosen through nodes N_1 and N_6. However, N_1 should not be chosen because it was used during the transmission of the first packet. N_6 is chosen as the next hop and, finally, the packet is forwarded to N_7, because N_{14} cannot relay the packet to the sink directly. In the same manner, the third packet will follow the path $N_2 \diamondsuit N_3 \ \diamondsuit N_8$, as depicted in Figure 7.

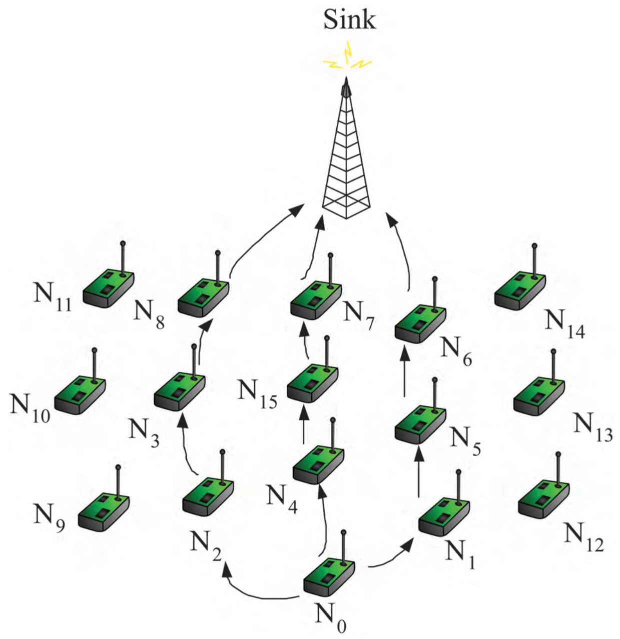

Figure 7. Sample scenario with LEERA.

3.3. Load balance routing with multiple sinks (LEERA-MS)

With respect to the origination of consequent packets from a single node, congestion will arise at the nodes residing in the middle of the topology. In parallel with the increasing packet intensity, the packets will suffer delays. Those delays are not the same as the back-off times in carrier sense multiple access with collision avoidance (CSMA/CA). Since a periodic sleep-wakeup schedule is employed in this project, although the back-off timer of a node expires, that node cannot capture the physical channel immediately. It should wait for the next wakeup period in order to send its data.

An alternative solution that we propose in this paper for that challenge is spreading the consequent emerging packets all over the topology as much as possible, which also means distributing the load over more nodes. However, a sufficient amount of spreading is constricted unless multiple sinks are used. If there is only one sink in the network, all of the subsequent packets try to follow a path around a certain line of sight. Therefore, the nodes positioned around that certain line are always employed in the forwarding process. Moreover, when a packet gets closer to the sink, it will compulsorily need to arrive at the node positioned around the sink. Hence, nodes residing around the sink are depleted of energy quickly. This situation is called the hot-spot problem, which is discussed in [16]. Data collected by the sensor nodes cannot be directly transmitted to the sink. Data must be relayed by multiple other interrelaying nodes in order to reach to the sink. Obviously, the nodes residing around the sink will always have to take charge during the relaying of the packets to the sink. Hence, those nodes located in the hot-spot area will quickly be depleted of energy and die. The death of those nodes does not only affect the nodes themselves or the data communication of the area in which they are located. Since they are the only ones needed to convey the packets arriving from other parts of the network to the sink, all communication can halt. This situation is clearly depicted in Figure 7. N_7 is the only last hop that can relay packets to the sink. Therefore, N_7 is the first node that will be depleted of energy. However, if multiple sinks are positioned in the topology, packets emerging from N_0 do not have to travel around a single line toward N_7. When multiple sinks are used, nodes trying to forward the packet can direct the packet to the most suitable sink in the topology. Hence, N_7 does not have to compulsorily perform as the last hop. Nodes N_{11}, N_8, and N_{14} can be employed as a last hop through a sink.

With LEERA-MS, the topology is divided vertically into grids. Every sink owns a grid area, and every node residing in this grid area takes up this grid's owner sink as the reference sink and makes geographical positional calculations according to that sink's coordinates. In LEERA-MS, while choosing the next hop, the energy levels are considered first. If all of the neighbors have the same amount of energy, this time the one that makes the largest angle toward the corresponding sink is chosen. By choosing a farther node, nodes located around the line of sight from the originating node toward its corresponding sink will be saved from possible congestion.

The angle between a node and the sink, as shown in Figure 8, is calculated according to the Cosine theorem:

$$\alpha = ArcCos\left[\frac{(x^2+y^2-z^2)}{2xy}\right]. \tag{3}$$

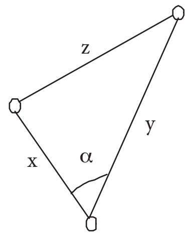

Figure 8. Calculation of angles.

During the transmission of the first packet, since all of N_0's neighbors have the same amount of residual energy, the first packet emerging from N_0 will follow the shortest path, i.e. $N_4 \diamondsuit N_1 \ \diamondsuit N_7$, as it is in LEERA. For the second packet, the same path is not used again. Since N_4 is employed in the forwarding process of the first packet, N_0's other neighbors should be used this time. This time, the one that has a larger angle toward the originator's reference sink is chosen as the next hop. Let us assume that N_1 has a greater angle toward the sink of the originator of the packet and is chosen as the next hop. The next hop for the second packet is chosen through nodes N_4, N_5, and N_{13}. Since N_4 is employed in the forwarding process of the first packet, it should be exempt this time. Again, there are 2 alternatives, N_5 and N_{13}. N_{13} has a larger angle toward the reference sink of N_0 than N_5 and is chosen as the next hop. The next hop of the second packet is chosen through nodes N_{14} and N_6. Since there are multiple sinks and N_{14} can directly relay the packet to its corresponding sink, it is chosen as the next and last hop. In the same manner, the third packet will follow the path $N_2 \diamondsuit N_9 \ \diamondsuit N_{10} \diamondsuit N_{11}$, as depicted in Figure 9.

3.4. Load balance routing with an energy level threshold (LEERA-TH)

Another alternative routing method that we propose is combining the LEERA method with a threshold mechanism in order to decrease the end-to-end delay. In this mechanism, a threshold's remaining energy level is predefined and the nodes can apply the shortest path method when choosing the next hop, until the closest neighbor to the sink consumes its energy and until the predefined threshold level is reached. If the sender node realizes that the closest neighbor has only as much as the threshold amount of remaining energy, it applies the energy-aware routing algorithm and chooses the next hop according to the LEERA-MS method.

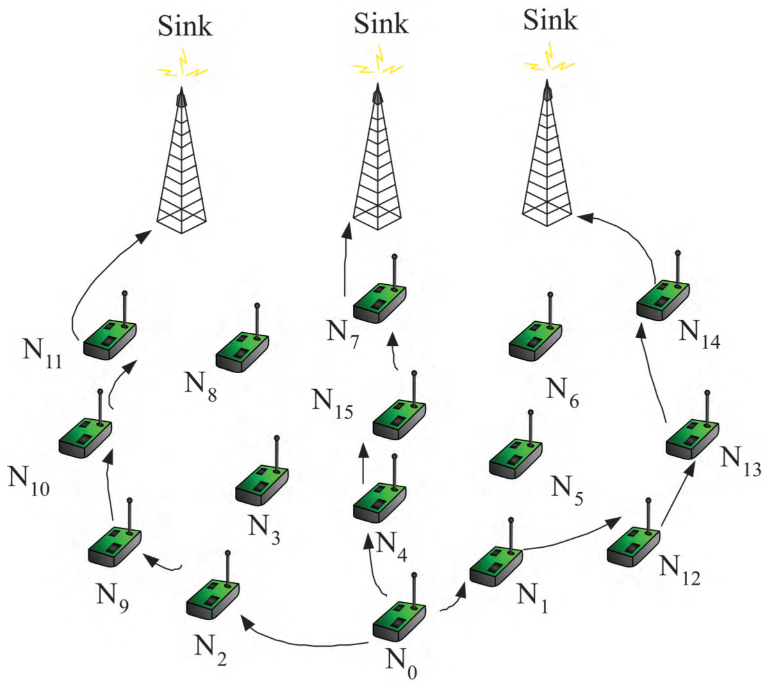

Figure 9. Sample scenario with LEERA-MS.

4. Performance evaluation

Simulations are performed on a testbed written in C++, and 802.11 CSMA/CA MAC is employed in order to avoid collisions. We use the following terms in our program:

T_d: Time period for data transmission (in our program, all of the data packets are assumed to be same length).

T_s: Setup latency (the difference between the time that a sender starts to send beacons for a specific receiver and the time at which it gets an acknowledgment from the receiver).

T_b: Time for every node to stay awake on f1 in order to determine whether any call for it is presented.

T: Time period for a node to wake up.

B_1: Transmit time of a beacon.

B_2: Interbeacon spacing.

We calculate the average T_s as in STEM:

$$T_s = (T + B_1 + B_2)/2. \tag{4}$$

Values assigned to the variables are:

$B_1 + B_2 = 150$ ms, $T_d = 4000$ ms, $T_b = 225$ ms, $T = 600$ ms.

We get characteristics of power consumption of the radio simulated from [17]:

$P_{Transmit} = 14.88$ mW, $P_{Receive} = 12.50$ mW, $P_{Idle} = 12.36$ mW, $P_{Sleep} = 0.016$ mW.

We ignore processing and other delays and accomplish the program by employing the topology with 176 nodes and 10 sinks. The network area is parceled into grids. Every node belongs to a grid and is thereby referenced to a sink depending on its geographical position.

Simulations are performed for a single scenario in which 50 packets that each take 0.2 ms to transmit from point to point emerge from N_0 with 0.1-ms intervals consequently.

Three different routing methods are simulated and compared with each other. The first is the shortest path algorithm, which employs the idea of choosing the neighbor closest to the sink. The second routing method is the localized energy-efficient (LEERA-MS) routing approach, which accomplishes forwarding according to the residual energy levels of the nodes in combination with their angles toward the packet originators' sink. Finally, the third method simulated is LEERA-TH, in which the threshold mechanism is combined with LEERA-MS. The total energy consumption in the network when applying the 3 different routing methods mentioned above is represented in Figure 10.

The amount of energy consumed by a single node and by the whole network for the duration of *LifeTime* is calculated as shown below.

$$E_{spent} = E_{wakeup} + E_{transmit} + E_{receive} \tag{5}$$

$$\begin{aligned} node[i].energyspent = {} & ((LifeTime/T) * T_b * P_{Idle}) + (T_d * P_{Transmit} * node[i].sndcounter) \\ & +(T_d * P_{Receive} * node[i].rcvcounter) + (T_s * P_{Transmit} * node[i].sndcounter) \\ & +((B_1 + B_2) * P_{Receive} * node[i].rcvcounter) + ((LifeTime/T) * (T - T_b) * P_{Sleep}). \end{aligned} \tag{6}$$

$$E_{total} = \sum_{i=1}^{n} Espent_i \tag{7}$$

As is clear in Figure 10, the total energy consumed by all of the nodes in the network is higher when the energy-aware routing algorithm is used as the routing method. At first glance, the result can seem strange to the reader; however, the idea in our approach is to prolong the lifetime of every node, thereby prolonging the lifetime of the network. Our algorithm causes more total energy to be consumed because the packets travel a longer way, since the nodes with more residual energy levels are chosen on the way to the sink. The shortest path algorithm seems to consume less energy than LEERA-MS. However, since the shortest path method always uses the same path for the same source-destination pair, the nodes on the path will quickly be depleted of energy. One way of healing the total energy consumption due to the increased distance to the sink is employing the LEERA-TH method. LEERA-TH does not consume as much energy as LEERA-MS because, up to a predefined critical threshold energy level, the shortest path algorithm is applied. When the amount of consumed energy by the nodes in the neighborhood of a sending node reaches the critical threshold value, then energy-aware routing is put into use. Thus, a better end-to-end transmission delay is achieved. Of course, there is always a trade-off between energy and delay.

Figure 11 shows the transmission time for transmitting all of the packets emerging from N_0 to the sink. Since the residual energy levels of the nodes are concerned during transmission between 2 nodes, rather than emphasizing closeness to the sink or to the sending node, the packets travel along more nodes and a longer path. This situation requires a longer time for the last packet to arrive to the sink, as shown in Figure 11. Since the node energy levels are not concerned with the shortest path algorithm, it takes a minimum amount of time for a packet to reach to the sink.

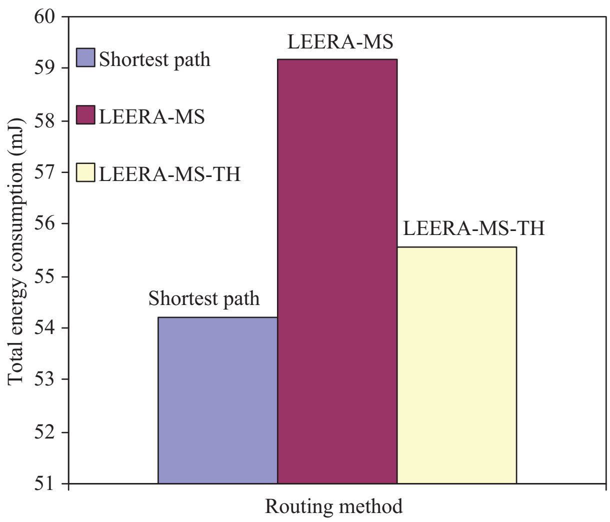

Figure 10. Comparison of total energy consumption in the network for each routing method.

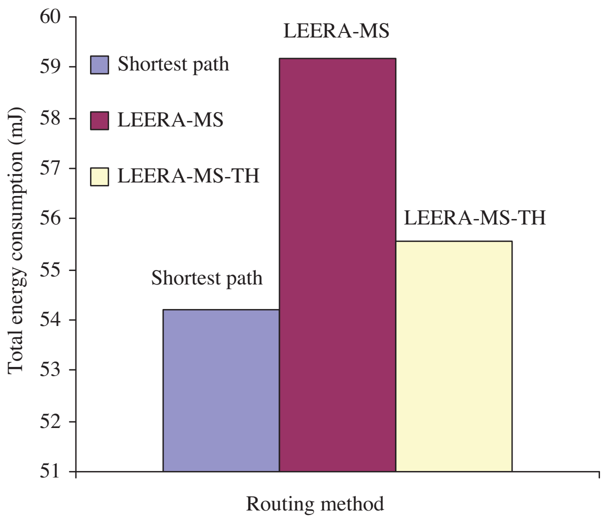

Figure 11. Transmission time.

Figure 12 represents the energy consumed by the node with the smallest residual energy level after all of the transmissions.

Obviously, energy consumed by the node with the least residual energy level is smaller for LEERA-MS and LEERA-TH than for the shortest path routing method. LEERA-MS outperforms other methods with a performance of about 40% better than the shortest path and 5% better than LEERA-TH in terms of network lifetime, which is obviously depicted in Figure 13.

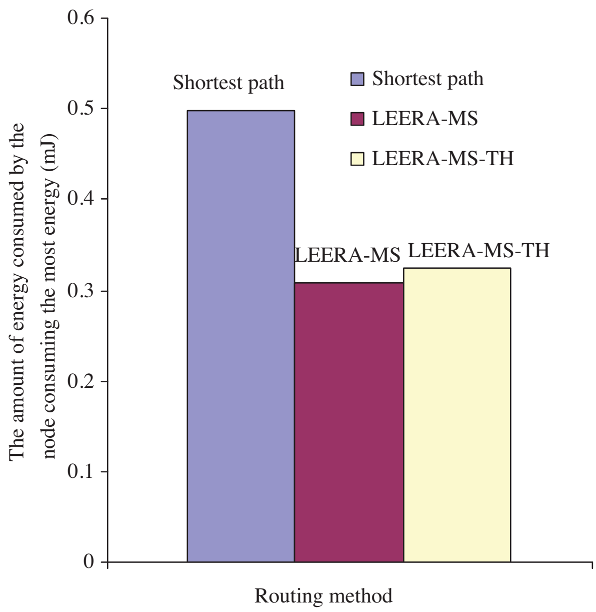

Figure 12. Comparison of energy consumption of the node with the smallest residual energy level.

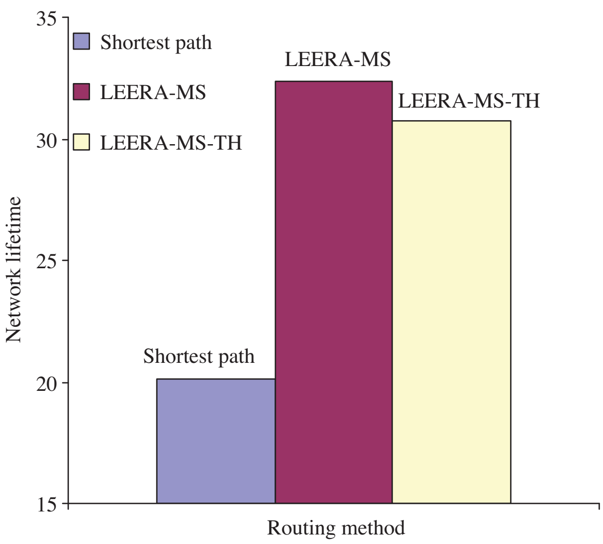

Figure 13. Comparison of network lifetimes.

5. Conclusion

In this paper, we proposed a localized power-aware routing method constructed over an energy-efficient sleep-wakeup schedule. Node energy levels in a network must be noticed by every node in the network. Accordingly, in recent studies, node energy levels have been broadcasted either in a neighborhood locally and periodically or in a flooding direction from the sink to every node in the network. Furthermore, some studies make energy cost estimates for the nodes in the network, which is unreliable. Obviously, transmission of energy level-informing data in the network brings too much overhead to the network. Moreover, nodes trying to convey this information consume energy, which is the main challenge in WSNs to be considered while designing a method. In our approach, the nodes already transmit RTS-CTS pairs to inform each other about a forthcoming transmission. Nodes do not perform an extra transmission to inform their neighbors about their residual energy levels. In addition, they do not have to make any cost estimation, which might not provide accurate data. In the method proposed here, the end-to-end delay increases because the same path is not always used for consecutive transmissions between the same source-destination pair. The path followed by the packets changes according to the residual energy levels of local nodes in the network. Accordingly, the number of nodes that take action in the forwarding process increases. Furthermore, as the number of forwarding nodes increases, total energy consumption of the network also increases. On the other hand, the main aim of WSN topology and protocol design is to prolong the lifetime of the networks. Prolonging the lifetime of the network can only be assured by distributing the load balance all over the network, which means using as many different paths as possible during a traffic flow. Numerical results of our method show that nodes live 40% longer than in a situation where the shortest path routing algorithm is used as the routing method. However, the end-to-end delay increases, which is not so vital unless a multimedia communication takes place. An alternative method can be to apply a threshold mechanism to our method. In this approach, nodes apply the shortest path routing method until they consume energy up to the critical threshold value. After the threshold level, the energy-aware routing method (LEERA-MS) can be applied in order to choose the next hop. This approach (LEERA-TH) decreases the end-to-end-delay, but it does not provide the energy efficiency supplied by LEERA.

In conclusion, there is always a trade-off between energy consumption and transmission time. For applications in which delay is not so vital, LEERA-MS provides the best performance in terms of the lifetime of the network.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, Vol. 40, pp. 102-114, 2002.

[2] M.M. Holland, R.G. Aures, W. Heinzelman, "Experimental investigation of radio performance in wireless sensor networks", Proceedings of the IEEE 2nd Workshop on Wireless Mesh Networks, pp. 140-150, 2006.

[3] H. Bahşi, A. Levi, "k-anonymity based framework for privacy preserving data collection in wireless sensor networks", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 18, pp. 241-272, 2010.

[4] G. Anastasi, M. Conti, M.D. Francesco, A. Passarella, "Energy conservation in wireless sensor networks: a survey", Ad Hoc Networks, Vol. 7, pp. 537-568, 2009.

[5] A. Savvides, C.C. Han, M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors", Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 166-179, 2001.

[6] S.M. Hedetniemi, S.T. Hedetniemi, A. Liestman, “A survey of gossiping and broadcasting in communication networks”, Networks, Vol. 18, pp. 319-349, 1988.

[7] B. Karp, H.T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks”, Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 243-254, 2000.

[8] Y. Wang, W.Z. Song, W. Wang, X.Y. Li, T.A. Dahlberg, “LEARN: localized energy aware restricted neighborhood routing for ad hoc networks”, Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, pp. 508-517, 2006.

[9] W.R. Heinzelman, J. Kulik, H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks”, Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 174-185, 1999.

[10] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, Proceedings of the Hawaii International Conference on System Sciences, pp. 1-10, 2000.

[11] S. Lindsey, C.S. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems”, Proceedings of the IEEE Aerospace Conference, pp. 924-935, 2002.

[12] R. Wu, M. Chen, Y. Su, H.J. Siddiqui, “A novel location-based routing algorithm for energy balance in wireless sensor networks”, Proceedings of WRI International Conference on Communications and Mobile Computing, Vol. 1, pp. 568-572, 2009.

[13] T. Cevik, D. Yiltas, A.H. Zalim, “A pipelining model in STEM protocol of sensor networks”, Proceedings of the 1st International Conference on Computing in Science & Engineering, pp. 42-46, 2010.

[14] T. Cevik, D. Yiltas, A.H. Zaim, “Delay efficient STEM by pipelining”, Procedia Computer Science, Vol. 3, pp. 96-103, 2011.

[15] C. Schurgers, V. Tsiatsis, M.B. Srivastava, “STEM: Topology management for energy efficient sensor networks”, Proceedings of the IEEE Aerospace Conference, Vol. 3, pp. 1099-1108, 2002.

[16] W. Yu, S.S. Wei, B.J. Shie, “Activating the hot spot problem by switching multiple sinks in wireless sensor networks”, Proceedings of the Fifth International Conference on Mobile Ad-hoc and Sensor Networks, pp. 204-211, 2009.

[17] X. Yang, N.H. Vaidya, “A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay”, Proceedings of Real-Time and Embedded Technology and Applications Symposium, pp. 19-26, 2004.