www.ietdl.org

Published in IET Wireless Sensor Systems
Received on 24th January 2012
Revised on 6th July 2012
doi: 10.1049/iet-wss.2012.0014

ISSN 2043-6386

Spreading-factor based addressing-tree protocol for ad-hoc wireless sensor networks

S. Saxena[1] *S. Mishra*[2] *A.K. Sharma*[3] *D.S. Chauhan*[4]

[1]*Department of Computer Application, Bharat Institute of Technology, Meerut, U.P., India*
[2]*Department of CSE, BTKIT, Dwarahat, Uttarakhand, India*
[3]*Department of CSE, YMCA, Faridabad, India*
[4]*Uttarakhand Technical University, Dehradun, Uttarakhand, India*
E-mail: sharadzoom@gmail.com

Abstract: Tree-routing (TR) is a fundamental protocol most suited for a tree-like sensor network. Here, a sensor node deployed as leaf node senses the environmental phenomenon and forwards it to the sink node positioned as the root node by following a fixed parent–child path. This strategy prevents the network from flooding path search messages and saves bandwidth. The enhanced-tree-routing (ETR) protocol is an improvement to the TR protocol that uses a structured node address assignment scheme and considers neighbouring tables stored at each sensor node to find the shortest path to the sink. This requires minimum computation energy, storage and provides energy-efficient routing. In this paper the authors have proposed a spreading factor-based addressing (SFBA-tree) approach based on a non-blocking orthogonal vector spreading factor addressing technique, for the TR protocol and usage of movable sinks to eliminate excessive multi-hopping caused by the ETR protocol while discovering the shortest paths. Also, the authors have shown an SFBA-tree implementation to the ZigBee networks to prevent storage space and computation. The simulation result shows that the new scheme provides energy-efficient addressing and communication in a moderate size tree network.

1 Introduction

An ad-hoc wireless sensor network consists of sensor-devices placed in a geographical environment in an *ad-hoc* way [1]. The network can be static or dynamic in nature. The sensors are tiny, self-configurable and adjustable entities with a short-range radio and processing unit powered by small portable batteries. Sensors are deployed to sense physical phenomena, such as sound, light, humidity, magnetic field, temperature and so on and to transmit it to a short range positioned collector called sink node. Sensors deployed near the sink can forward data using single-hopping, whereas sensors at remote distance have to follow a multi-hopping sequence to forward data. Furthermore, a wireless sensor network (WSN) is characterised as self-organising, as it is adaptable to sensor failure, degradation and can react according to task changes. They are utilised in various application areas such as battlefield monitoring, environmental study, animal tracking, oil and gas refineries and chemical detection and so on. Although, WSNs are found to be useful and have wide application areas, they are limited to their life cycle. All sensor activities such as sensing, computing and transmitting need power consumption supplied by small portable batteries. Manually replacing or recharging batteries of deployed sensors is an extremely challenging task. Therefore solutions to optimise sensors' lifetime are important. Moreover, every aspect of design, deployment and management of a WSN has to be energy-efficient to meet typical power requirements [2]. As dictated by the task requirements in [3] radio component in a sensor consumes comparatively more power than the other components. This is because of two reasons, first, the transmit power of a wireless radio is proportional to the distance squared; second, a single-hop direct communication over long distance consumes more energy than multi-hop communication. Multi-hop transmission is, therefore, considered to be a more energy-efficient method in most of the applications. Topology creation and routing for multi-hopping thus become an essential function of the WSN. Routing is the method available in the firmware of each sensor node to find the paths between source and destinations.

The elementary method of the sensor network construction is to start with a root node (usually sink), expand as new nodes and join as child nodes. Each node can have multiple children but only one parent. The resultant network structure is like a tree as depicted in Fig. 1. In Fig. 1, nodes A, B and C are the child nodes of the root node. Both root and B are the ancestors of the nodes E and F, whereas all the nodes except the root are descendant nodes.

This paper is organised as follows: Section 2 reviews the related work. Section 3 presents the proposed SFBA-tree protocol. Section 4, shows the hop-count and energy comparison of protocol. Section 5 describes SFBA-tree implementation to the ZigBee networks. Performance of SFBA-tree in ZigBee is shown in Section 6. Finally, Section 7 concludes the paper.

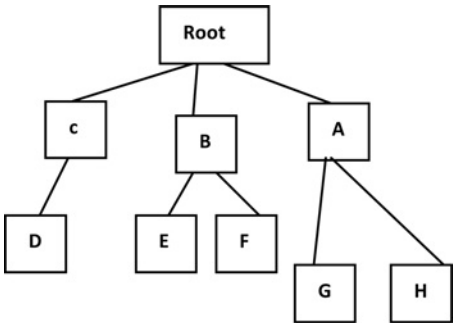

Fig. 1 *Basic tree topology*

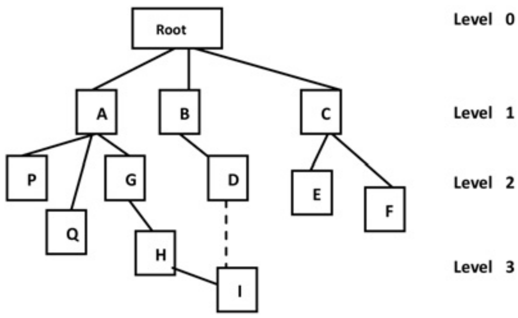

Fig. 2 *ETR-tree topology*

2 Related work

Data transmission in WSNs is different from the traditional wired and wireless communication based on transmission control protocol (TCP)/internet protocol (IP) and wireless access protocol (WAP) architecture. Since sensors are limited with their processing and storing capacity, heavy computations and calculations are not permitted. A different network topology and protocol schemes are therefore needed. The authors in [4] discussed time division multiple access (TDMA)-based protocols, where sensor nodes turn on their radio in their allotted time slots and sleep for the rest of the time. TDMA thus helps in saving energy in idle time and eliminates problems of interference. Data-centric routing discussed in [5] is based on the querying concept. The node desiring certain types of information sends queries to the identified regions and waits for reply messages from selected regions. During wait time the devices can switch off the radio to save energy. Clustering based approaches discussed in [6] group nodes into clusters. Cluster head in a cluster is nominated for all intra-cluster data collection, communication and inter-cluster transmission in order to save energy. Location-based protocols take advantage of location information to increase the energy efficiency in routing by relaying the data to the desired regions rather than the whole network [7]. *Ad-hoc* on demand distance vector (AODV) protocol finds the shortest paths based on hop-count metrics and forwards the data through the identified route [8]. For this purpose, whenever a node requires a route, it initiates a route discovery procedure broadcasting route request messages. Algorithms that search for alternatives to the parent–child links have recently been proposed specifically for the ZigBee networks [9].

Tree routing (TR) is a simple and efficient protocol for a moderate tree-like network (Fig. 1). The protocol follows only parent–child links starting from root node to leaf node [10]. TR thus eliminates path searching and updating complexities. TR is suitable for networks consisting of small-memory, low-power and low-complexity lightweight nodes. The main drawback of TR is the increased hop-count as compared to other path search protocols. TR does not utilise the neighbour table fully. A neighbour table records information such as, addresses of nodes within the radio range, information of parent and child nodes and so on. The neighbour table is created when the node joins a parent node. Enhanced tree routing (ETR) assumes that each node has an updated neighbour table having the address of its immediate one-hop neighbour [10]. This neighbour table is utilised to identify the alternate shortest path to the sink node with a hop-count less than the actual path. For peer-to-peer communication each node has a unique identification number. This number is assigned to the node when it joins the network. Fig. 2 shows the architecture of the ETR protocol. Here, node I will select the path (I, D, B to root, having hop-count 3) instead of the traditional path (I, H, G, A to root, having hop-count 4). Further, a new non-blocking orthogonal variable spreading factor with time multiplexing (NOVSF-TM) based addressing to a sensor node and topology comparison of the three protocols, that is, TR, ETR and NOVSF-TM-based addressing protocol is shown in [11]. The proposed scheme reduces the address length and involves minimum computation. The excessive multi-hopping caused by the ETR is eliminated by the NOVSF-TM addressing technique.

3 SFBA-tree protocol

The NOVSF-TM technique is used for contention free channel allocation among the contenders [12]. This scheme utilises timely shared orthogonal codes by the number of channels in the wireless-code division multiple access (W-CDMA) system. In our proposed work, we have planned to use these unique codes to represent the sub-region in a large two-dimensional sensor network. A sub-region is a small part of the large geographical area of sensor node deployments. Each sub-region is hosted by a movable sink, placed at the central place of the region called a centroid. The centroid location of the sub-region is obtained by logically joining the extreme sensors as coordinates so as to form a polygon. The movement of the sink is limited to repositioning it within the sub-region to eliminate excessive hop-counts created in the ETR protocols. The orthogonal code of the movable sink is used to identify the sub-region as well as the sinks uniquely. Every sensor in the sub-region connects to the movable sink either through a single-hop or minimum multi-hop distance. The elementary architecture of the protocol having one movable sink and a fixed sink node is shown in Fig. 3. The movable sink of the sub-region provides address to the sensors in the sub-region by combining its code and a sequence number, to uniquely identify the sensor node (Fig. 5 and Table 1). For analysis, we have utilised the SF-8 NOVSF-TM code (Fig. 4) for address assignment to the movable sink. Overall four movable sinks with orthogonal codes (as address) MS1 = 1111, MS2 = 11-1-1, MS3 = 1-11-1 and MS4 = 1-1-11 can be positioned in the region with SF-8. The template for a sensor node address is shown in Fig. 5.

The network address (e.g. for MS1 = 1111 1111 0 000 0000) in any range, has last seven bits as zero, which is the address of a movable sink in the network. The other device in the network has bits other than '0', which means only the movable sink in the network can have '0' bits in the

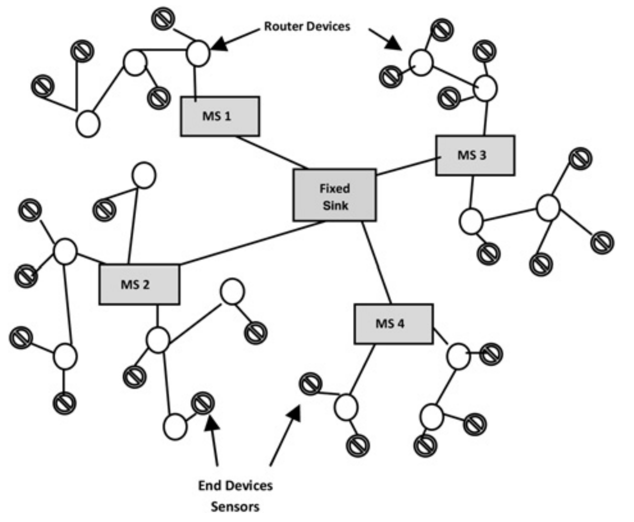

Fig. 3 *SFBA-tree topology with fixed sink node and moveable sinks*

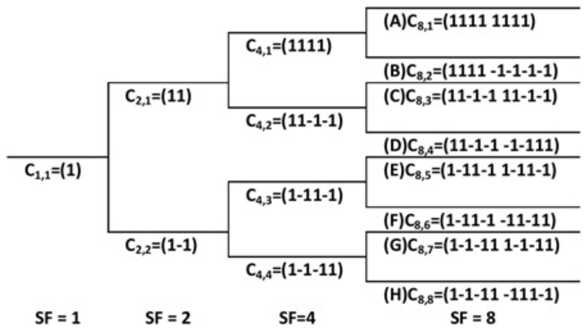

Fig. 4 *NOVSF-TM code tree structure (SF 8)*

Sensor Node Address

Moveable Sink Address (8 bit)		Sensor Number	
1111	1111	**0 000**	0001

Fig. 5 *SFBA-tree node addressing template and example*

address and the other device has non-zero bits. The ninth-bit in the address can take two values either '0' or '1' indicating either for a coordinator node or end-device, respectively. Bits '10–12' are used to store the depth of the movable sink from the root node. A maximum seven level 0–7 are supported by a single code. The last four digits contain the sequence number from 1 to 15 as the unique identity for the end-device (sensors). The centroid location determination algorithm is shown in Fig. 6. In Fig. 6, $X(N)$ and $Y(N)$ are the two linear arrays to hold the random position of sensor nodes as coordinates (X, Y). The *T_Range* parameter holds the transmitter range of a sensor node. The topology is created by the random placement of a sensor at some coordinate position determined by *rand*() function. The maximum limit of the sensor is depicted by the variable N. The centroid point is determined by (1).

$$\text{Centroid}\,(p, q) = \Big[1/6A \sum (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),\; 1/6A \sum (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)\Big] \tag{1}$$

where $A = 1/2\sum(x_i y_{i+1} - x_{i+1} y_i)$, for all $i = 0$ to $N - 1$.

$$\text{dist} = \text{sqrt}((x_2 - x_1)^2 + (y_2 - y_1)^2) \tag{2}$$

Sensors are now connected to the movable sink by using a distance formula stated in (2). The algorithm finally generates a dynamic tree topology with a fixed sink node and movable sink in that region.

4 Energy and hop-count comparison of protocols

We have conducted simulations in an event-driven simulator developed in MATLAB to compare the performance of TR, ETR and SFBA-tree in terms of hop-count and energy consumption. For analysis, we have assumed a small region of $500 \times 500\,\text{m}^2$. A fixed sink node is available for data collection. After the sensors are randomly deployed in the region, a logical polygon is created with the sensor nodes at an extreme location, as the coordinates of the polygon. The centroid is the central position of the polygon that can cover maximum number of sensors. With SF-8 the architecture can support a maximum of four movable stations in the region. An orthogonal code can support addresses in the range (0001 to 1111). Initially one movable sink is placed in the region. As the number of sensors increases beyond 30 (15 + 15), a second movable sink is positioned, if it increases beyond 60 the third movable sink is positioned and after 90, the third movable sink is placed and finally, the fourth is placed to support a maximum of 120 sensors. Hence, a tree-like structure is obtained as shown in Fig. 7. The sensors send their data to the movable sink either in single-hop or multi-hop manner. The various simulation parameters are shown in Table 2.

We have generated some dynamic network topologies and tested the three protocols on it. In particular, after the nodes are deployed, the coordinator is powered on to start the

Table 1 Sensor node addressing

Sl no.	Movable sink	Movable sink address	Orthogonal codes for addressing	Generated node addresses
1	MS1	1111	A = 1111 1111	1111 1111 0 000 0001 to 1111 1111 1111 1111
			B = 1111-1-1-1-1	1111 -1-1-1-1 0 000 0001 to 1111 -1-1-1-1 1111 1111
2	MS2	11-1-1	C = 11-1-1 11-1-1	11-1-1 11-1-1 0 000 0001 to 11-1-1 11-1-1 1111 1111
			D = 11-1-1-1-111	11-1-1 -1-111 0 000 0001 to 11-1-1 -1-111 1111 1111
3	MS3	1-11-1	E = 1-11-1 1-11-1	1-11-1 1-11-1 0 000 0001 to1-11-1 1-11-1 1111 1111
			F = 1-11-1-11-11	1-11-1 -11-11 0 000 0001 to 1-11-1 -11-11 1111 1111
4	MS4	1-1-11	G = 1-1-11 1-1-11	1-1-11 1-1-11 0 000 0001 to 1-1-11 1-1-11 1111 1111
			H = 1-1-11-111-1	1-1-11 -111-1 0 000 0001 to 1-1-11 -111-1 1111 1111

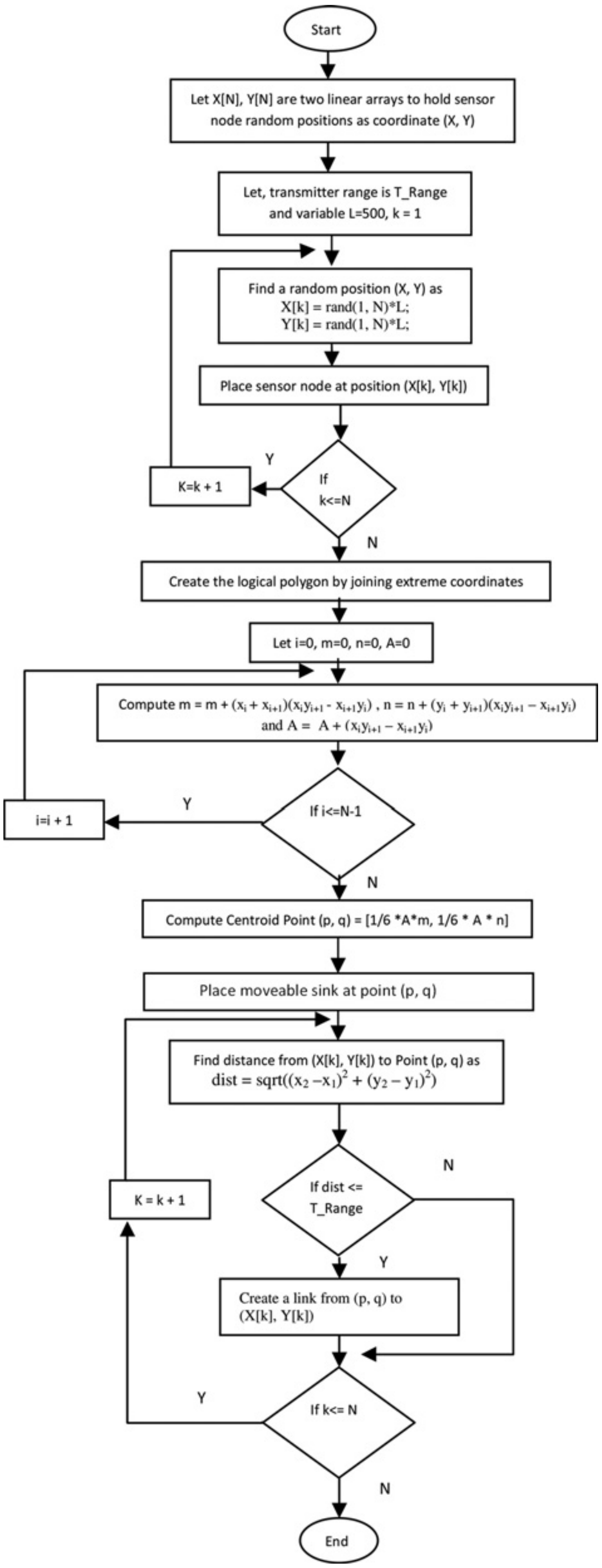

Fig. 6 *Movable sink positioning algorithm*

network. All the nodes then power on and search their neighbourhood for parents. The new node and its identified parent exchange joining information and a network address are assigned to the new node. The network is established when all the nodes join the network. An event is a transmission of packets from a randomly selected source node to a randomly selected destination node by using a route determined by the three protocols. For each event, the

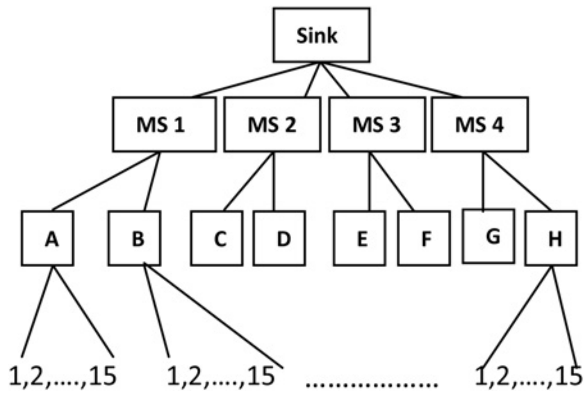

Fig. 7 *SFBA-tree architecture*

number of hops and the energy consumption of each hop are recorded. There is a sequential execution of events that is, the second event triggers only when the first one finishes. Therefore there is no channel contention or packet loss during transmission. This factor makes us focus only on the performance of protocols by considering only the hop-counts and energy consumption in the route generated. The energy consumption model specified in [13] is used; according to which the energy required by a single-hop transmission of a packet is $(0.001 \times d^3)$, where d is the distance between the two nodes. For each network simulation scenario, NWKS = (30, 32 40, 45, 50, 60 and 65) instances of sensor networks are randomly generated and RUNS = 10 000 runs are conducted for each instance. For each instance, the hop-count and energy consumption are recorded. The results of network instances are average to find the metrics (3) and (4)

$$\mathrm{AvgHops} = \frac{1}{\mathrm{NWKS} * \mathrm{RUNS}} \sum_{i=1}^{\mathrm{NWKS}} \sum_{r=1}^{\mathrm{RUNS}} h_{r.i} \tag{3}$$

$$\mathrm{AvgEng} = \frac{1}{\mathrm{NWKS} * \mathrm{RUNS}} \sum_{i=1}^{\mathrm{NWKS}} \sum_{r=1}^{\mathrm{RUNS}} e_{r.i} \tag{4}$$

where $h_{r.i}$ and $e_{r.i}$ are the hop-count and energy-consumption of the rth run for ith network instance, respectively.

We have considered two cases for the simulation:

Case 1: The transmitter range is set to 235 m and the number of nodes deployed are taken in the range (20, 25, 32, 40, 45, 50, 60 and 65). The simulation results are shown in Figs. 8 and 9. It is clear from Fig. 8 that for randomly selected sensors, the TR, ETR and SFBA-tree show a noticeable difference in hop-count to the sink node. The TR protocol shows a high line of hop-count as it follows strict parent–child path to the sink node. The ETR shows comparatively

Table 2 Simulation parameters

Simulation parameters	
region	500×500 m^2
sensor nodes	in range (30, 32 40, 45, 50, 60 and 65)
initial energy	0.5 J
packet length	200 bits
feasible site identified	15
transmitter range	230 m
1 round	equal to 100 time frame
β	0.2

IET Wirel. Sens. Syst., 2012, Vol. 2, Iss. 4, pp. 369–376
doi: 10.1049/iet-wss.2012.0014

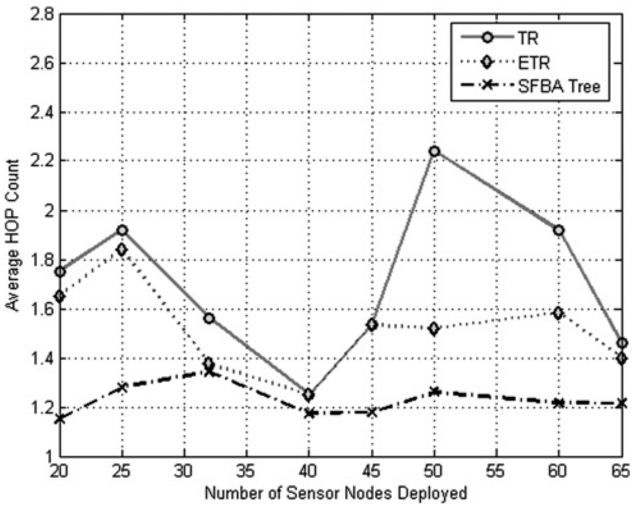

Fig. 8 *HOP count comparison of TR, ETR and SFBA protocols in case 1*

less number of hop-counts to the TR protocol because it considers a neighbour table to select the shortest path to reach the sink node. Finally, the SFBA-tree has the lowest hop-count. The hop-count reduction in it is observed because of the positioning of movable sinks to the centroid location of the region and elimination of excessive hop-counts as most of the sensors are now directly connected to the movable sink to send or receive data. For a moderate size network of nodes 40 or 45 the TR and ETR protocols show the same number of hop-counts. Movable sinks accumulate the data and forward it to the fixed sink node.

The energy consumption in transmitting a single data packet is dependent on the distance between the two adjacent nodes. A small distance has less energy consumption as compared with a large distance. Based on distance matrices, the energy consumption of three protocols is shown in Fig. 9. It is evident from Fig. 9 that the energy consumption reduces slowly to a certain point, as more number of sensors are deployed. This is because of multi-hopping of data packets using small paths. The energy consumption increases thereafter, because the excess

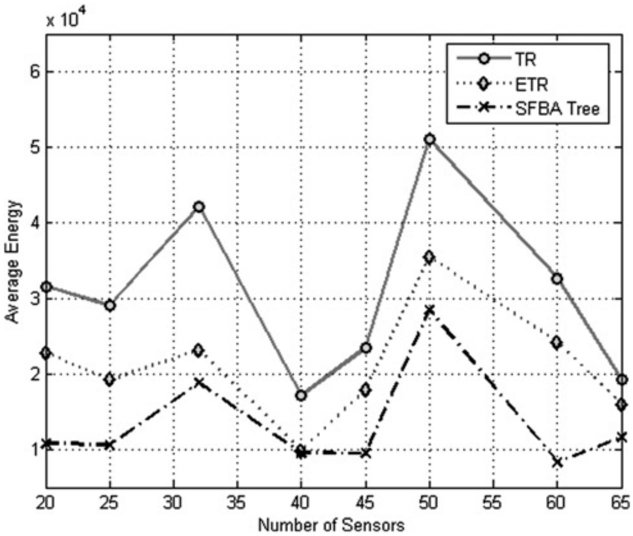

Fig. 9 *Average energy comparison of TR, ETR and SFBA tree in case 1*

in hop-counts outweighs any possible decrease in single-hop distances. In practice, dense deployment is used not for energy efficiency. Rather, it is used for providing the required measurement density and the radio connectivity redundancy needed to deal with issues such as node failure and so on. Therefore from Fig. 9, it is evident that the SFBA-tree reduces more energy than the TR and ETR protocols.

Case 2: The number of nodes deployed is fixed to 40 (one selected value from the range in Case 1), whereas the maximum radio range is taken in the range (230, 235, 240, 245, 250, 255 and 260). The simulation results are shown in Figs. 10 and 11. Fig. 10 shows that at the transmitter range 230 and 235, the ETR protocol performs better than the other two protocols, because the number of nodes moderate and no dense deployment exists. As the transmitter range increases, the coverage area of the sensors increases and thus hop-counts are tending to decrease as more number of sensors can either directly attach to the sink node or to the movable sink. For ETR, a large radio range provides more number of neighbours and hence, availability of more number of alternative shortest paths. In the SFBA-tree, the increase in the transmitter range causes

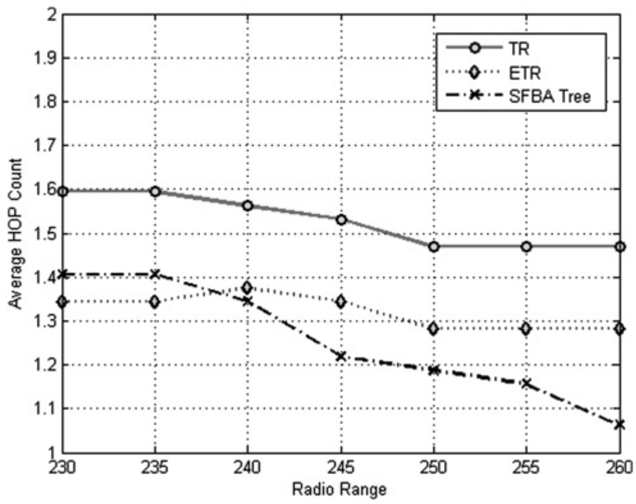

Fig. 10 *Average HOP count comparison of TR, ETR and SFBA tree in case 2*

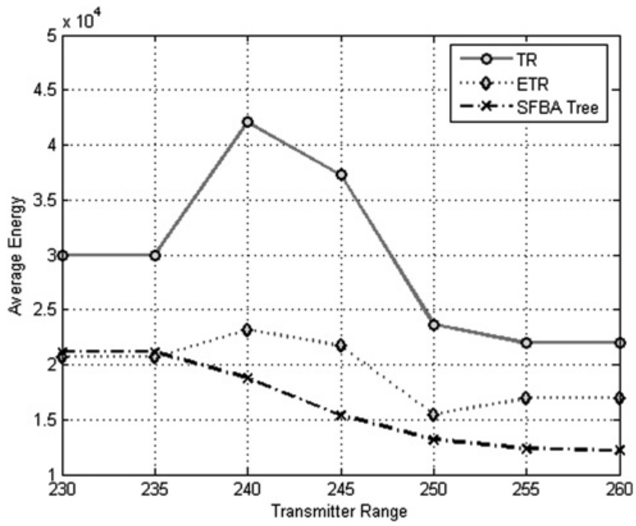

Fig. 11 *Average energy comparison of TR, ETR and SFBA tree in case 2*

direct attachment of sensors to the movable sink. This leads to reduction of multi-hopping to single-hopping and hence, reduction in the hop-count.

As for energy consumption, it is clear from Fig. 11 that for a transmitter range 230 and 235 both the ETR- and SFBA-tree perform almost same, but with increase in transmitter range, the energy consumption in both TR and ETR protocols is higher than the SFBA-tree because of the increase in per hop distance. The movable sink positioning in the SFBA-tree with increased transmitter range causes direct attachment of sensors to the sink node and hence, multi-hopping is eliminated. Thus, the distance becomes shorter and energy consumption is reduced. Therefore the SFBA-tree is more energy-efficient than the two protocols.

Furthermore, other factors that cause variation in the plots obtained in Figs. 8–11 are owing to receiver sensitivity, transmit-power values, deployment field size, radio range, interferences and the node numbers. These parts of the work are kept reserved for further studies.

5 SFBA-tree implementation to the ZigBee network

ZigBee is a standard for *ad-hoc* networks that is based on IEEE 802.15.4 [14, 15]. It has two routing options, first is similar to the AODV protocol and second is the TR protocol. The ETR protocol makes an improvement to it and uses path indexes for routing information and uses less memory space. The root node in ZigBee is the coordinator called router node. The other nodes can be router node or end devices. A router node can further accommodate child nodes, whereas an end device cannot have any children of its own. Every node in ZigBee has a pre-configured globally unique 64-bit identity called the IEEE address. When a node is joined as child node, it gets a 16-bit network address from its parent. The network address is based on a hierarchical tree structure and is computed by the parent node based on its own network address and the network address of its current children. The '0' address is reserved by the coordinator and the non-coordinator takes only a non-zero network address. In a region, the nodes communicate by their unique network address only, leading to reduced packet length. The IEEE address of a node cannot be changed but the network address is changeable as it moves from one region to another. In this section, the network address structure of ZigBee will be analysed and utilised to implement the tree protocol. The network address in ZigBee is calculated by some network parameters set (C_m, R_m, L_m), which is configured to be same for all the nodes in the network.

C_m: maximum allowed children to a parent;
R_m: maximum router children a parent can have;
L_m: maximum depth of the network.

In short, the network address in ZigBee is computed as: based on parameters set (C_m, R_m, L_m) and network depth d each router computes $C_{skip}(d)$, which is the address block size of its router child (see (5)).

If the router addresses are numbered from 1 to R_m and the end device addresses are numbered from $(R_m + 1)$ to C_m a parent node at depth d with network address A_p assigns the following address to its children

$$A_c = \begin{cases} A_p + C_{skip}(d)(k-1) + 1 & (1 \le k \le R_m) \quad (a) \\ A_p + C_{skip}(d)R_m + k - R_m, & R_m < k \le C_m \quad (b) \end{cases} \tag{6}$$

The address assignment scheme given in (5), 6(a, b) can be applied readily to the tree structure shown in Fig. 1. Furthermore, for a node at depth d with address A_m which determines the next hop for the destination address A_d the following statement holds:

1. If $A_m < A_d < A_m + C_{skip}(d-1)$, then A_d is the descendant of A_m.
2. If $A_d > A_m + R_m \times C_{skip}(d)$, then A_d is an end device child of A_m.
3. The next hop to a descendant which is not an end device is given by

$$A_n = A_m + 1 + \left\lfloor \frac{A_d - A_m - 1}{C_{skip}(d)} \right\rfloor C_{skip}(d) \tag{7}$$

In order to apply the SFBA-tree to the ZigBee network, we need to understand the address pattern of the coordinator and the end-device in the network. The address format for a 16-bit network address is shown in Fig. 12. The first eight bits in the address field are the unique NOVSF code assigned to a coordinator in a tree hierarchy, the ninth bit in the address is called flag bit(f) and can have only two values 0 or 1. The value 0 indicates the address of a coordinator and 1 indicates the address of an end device. The bits 10–12 hold the depth ('level') of a node and bits 13–16 hold the unique identity assigned by the coordinator to a device. The depth of highest depth common ancestor (HDCA) for an SFBA-tree can be determined by the 'Tree Level' ('bits 10, 11, 12') from the network address. For two nodes, N_a and N_b, if $l_i^a = l_i^b$ $(i = 1, \ldots, k)$ and $1 <= k <= L_m - 1$, then k will be their HDCA, otherwise the coordinator will be their HDCA at depth 0. Furthermore, the following conditions are found:

Condition 1: N_b is the descendant of N_a, if $d_b > d_a$ and $l_i^a = l_i^b$ $(i = 1, \ldots, d_a)$;
Condition 2: N_b is the grandchild of N_a, if $d_b = d_a + 2$ and $l_i^a = l_i^b$ $(i = 1, \ldots, d_a)$;
Condition 3: N_b and N_a are siblings, if $d_b = d_a$;
Condition 4: N_b is the child of N_a, if $d_b = d_a + 1$ and $l_i^a = l_i^b$ $(i = 1, \ldots, d_a)$.

Here, d_a and d_b are the depth of the two nodes N_a and N_b from the coordinator node. The maximum value for the network depth in the ZigBee standard is 7 that can be stored easily in the 'Tree Level' parameter of the network address. If $C_m = 16$, devices can be connected to a coordinator, a four-bit 'Device ID' is sufficient to store it. Hence, 28 bits are required to store a single-tree structure. If a complete 16 bit address is used as in [16, 17] to store the tree structure,

$$C_{skip}(d) = \begin{cases} 1 + C_m(L_m - d - 1), & R_m = 1 \\ \dfrac{1 + C_m - R_m - C_m R_m^{L_m - d - 1}}{1 - R_m}, & \text{otherwise} \end{cases} \quad d = 0, 1, \ldots, (L_m - 1) \tag{5}$$

doi: 10.1049/iet-wss.2012.0014

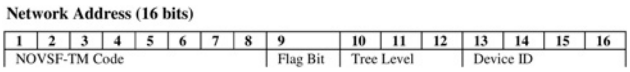

Fig. 12 *Network address for a device in ZigBee*

we need $6 \times 16 = 96$ bits (for 6 ancestors). The cost of this type of addressing is negligible as the address assignment task is performed by the coordinators at the startup of the network and no additional computation cost is required.

6 Performance analysis

In this section we conduct simulation in MATLAB to evaluate the performance of the SFBA-tree against the traditional TR protocol available in the ZigBee network. The two protocols are compared in terms of hop-count and energy consumption. The ZigBee parameter used is $(C_m, R_m, L_m) = (5, 2, 5)$. The network is shown in Fig. 13. For an address assignment scheme the ZigBee standard is followed. When all the nodes are physically deployed in the location, the coordinator is powered on to start the network. The other node is powered on and searches its neighbours for a router that can accommodate more children. The joining node and the parent exchanges joining request and reply messages to complete the network. An event is transmission of a network packet from a source node to a destination along the route determined by the TR- and SFBA-tree protocol. For each event, the number of hops and the energy consumption of each hop are recorded. The events are executed sequentially one after another. At each run, a source and destination pair is randomly selected and a hop-count for the source-to-destination transmission (h_r) is recorded. RUNS = 10 000 runs are conducted for each case and the average hop-count is used as metric (8)

$$\text{Average hops} = \frac{1}{\text{RUNS}} \sum_{i=1}^{\text{RUNS}} h_r \tag{8}$$

Assuming all nodes have same transmission range T_r, by varying the range of it like (1.10, 1.20, 1.50, 1.80)R we can obtain different network topologies. It is assumed that there is no packet loss in these cases. The average hop-count using (8) is computed. The graph is shown in Fig. 14. When the transmitter range is 1.10R no node is in the transmitter range of the other. Hence, no possibility of shortest path exists. Therefore the traditional parent–child route is followed. As the transmitter range increases, more nodes come in the proximity of the other node. The SFBA-tree then starts using the neighbour table to find the shortest path to the sink. Hence, at transmitter range 1.80R, the two protocols show a noticeable difference in the hop-count. For power the consumption model in [16] is used; according to which

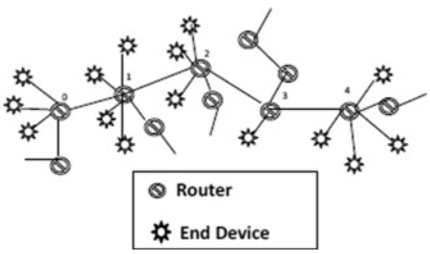

Fig. 13 *ZigBee network with (5, 2, 5)*

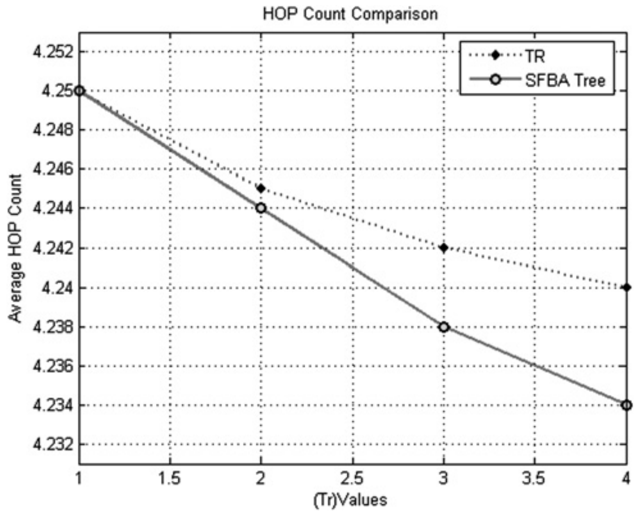

Fig. 14 *HOP count comparison of TR- and SFBA-tree protocols in the ZigBee network*

transmitting a k bit packet over a distance d metres is given by

$$E_r(k, d) = E_{elec}.k + \varepsilon_{amp} k.d^3 \tag{9}$$

where $E_{elec} = 50$ nj/bit is the energy dissipated by the transmitter circuitry. $\varepsilon_{amp} = 100$ pj/bit/m^3 is the energy dissipated by the transmitter–amplifier to achieve an acceptable signal-to-noise ratio. Moreover, the energy cost of a route can be given by

$$E = 2E_{elec}kM + \varepsilon_{amp}k. \sum_{m=1}^{M} d^3 \tag{10}$$

where M and d_m are the hop-count of the route and the transmission distance of the mth hop, respectively. For Fig. 14 a network scenario of $R = 100$ m and $k = 10$ Bytes is taken. For each run the energy consumption (E_c) for a randomly chosen source–destination pair is recorded. The average energy consumption for RUNS = 10 000 is

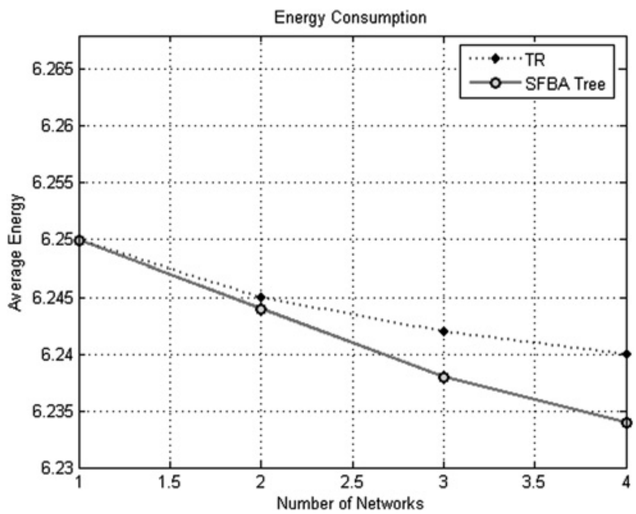

Fig. 15 *Energy consumption comparison of TR-and SFBA-tree protocols in ZigBee*

computed by the matrix

$$\text{AverageEng} = \frac{1}{\text{RUNS}} \sum_{c=1}^{\text{RUNS}} E \tag{11}$$

The simulation result is shown in Fig. 15. For an energy consumption comparison, the four transmitter ranges are used to create four different network topologies. The energy, therefore, is recorded with the help of (11). For transmitter range 1.10R, both the protocols show same energy consumption, because no short-cut path is utilised by the ETR protocol because of the unavailability of the other node in the transmitter range. As the radio range goes on increasing, alternate paths are identified with the help of a neighbour table. Hence, a significant difference is observed in the two protocols in terms of energy consumption.

7 Conclusion

In this paper, we have exposed an improved addressing and routing strategy over the two existing protocol called TR and ETR. The TR protocol, being simple and less complex is suitable for small sensor networks, but it does not utilise the neighbour table for link optimisation. The ETR protocol makes use of these alternative links available in the neighbour table to optimise the routing paths. The ETR becomes complex when the density of the sensor nodes increases. The SFBA-tree uses orthogonal codes as addresses to the sensor nodes. The sensor utilises this orthogonal code as node address and uses them as their network address. The positioning of the movable sink in the region at the centroid causes reduction in excessive hop-count occurring in the ETR protocol. Furthermore, we have shown a new addressing scheme implementation to the ZigBee network that shows significant effort in reducing energy in terms of address storage and referencing. The SFBA-tree technique is found to be more energy-efficient and easy to implement. The simulation results show that the SFBA-tree can outperform ETR and TR in terms of hop-count and energy.

8 References

1 Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: ‘A survey on sensor networks’, *IEEE Commun. Mag.*, 2002, **40**, (8), pp. 102–104

2 Raghunathan, V., Schurghers, C., Park, S., Srivastava, M.: ‘Energy-aware wireless micro sensor networks’, *IEEE Signal Process. Mag.*, 2002, **19**, (2), pp. 40–50

3 Zhao, F., Guibas, L.: ‘An information processing approach, wireless sensor networks’ (Elsevier–Morgan Kaufmann, Boston, 2004)

4 Rhee, I., Warrier, A., Aia, M., Min, J.: ‘Z-MAC: a hybrid MAC for wireless sensor networks’. Proc. ACM SenSys, S Diego, USA, 2005

5 Intanagonwiwat, C., Govindan, R., Estrin, D.: ‘Directed diffusion: a scalable and robust communication paradigm for sensor networks’. Proc. Sixth Annual Int. Conf. on Mobile Computing and Networks (MobiCom 2000), Boston, 2000, pp. 56–67

6 Heinzelman, W., Chandrakasan, A., Balsakrishnan, H.: ‘Energy-efficient communication protocol for wireless sensor networks’. Proc. Hawaii Int. Conf. System Sciences, Hawaii, 2000, p. 8020

7 Sohrabi, K., Gao, J., Ailawadhi, V., *et al.*: ‘Protocols for self-organization of a wireless sensor network’, *IEEE Pers. Commun.*, 2000, **7**, (5), pp. 16–27

8 Perkins, C.E., Royer, E.M.: ‘Ad hoc on-demand distance vector routing’. Proc. Second IEEE Workshop Mobile Computing Systems and Applications, 1999, pp. 90–100

9 Taehong, K., Daeyoung, K., Noseong, P., Seong-eun, Y., Lopez, T.S.: ‘Shortcut tree routing in ZigBee networks’. Proc. Second Int. Symp. on Wireless Pervasive Computing, 2007

10 Qiu, W., Skafidas, E., Hao, P.: ‘Enhanced tree routing for wireless sensor networks’ (Elsevier-Ad hoc Networks, 2009), pp. 638–650

11 Saxena, S., Mishra, S., Sharma, A.K., Chauhan, D.S.: ‘Non-orthogonal variable spreading factor-time multiplexing based sensor network topology compared with tree routing and enhance tree routing protocol’, *Int. J. Comput. Appl.*, 2011, **31**, (2), pp. 40–44

12 Vadde, K., Cam, H.: ‘A code assignment algorithm for non-blocking OVSF codes in WCDMA’ ‘Telecommunication systems’ (Kluwer Academic Publisher, Netherland, 2004), pp. 417–431

13 Park, J., Sahni, S.: ‘An online heuristic for maximum lifetime routing in wireless sensor networks’, *IEEE Trans. Comput.*, 2006, **55**, (8), pp. 1048–1056

14 ZigBee Alliance, ‘ZigBee Specification Version 1.0’, 2005

15 Institute of Electrical and Electronics Engineers, Inc.: ‘IEEE Std. 802.15.4–2003, IEEE standard for information technology’ (IEEE Press, New York, 2003)

16 Zhang, H., Shen, H., Tan, Y.: ‘Optimal energy balanced data gathering in wireless sensor networks’. Proc. IEEE Int. Symp. on Parallel and Distributed Processing, 2007

17 Perkins, C.E., Royer, E.M.: ‘Ad-hoc on-demand distance vector routing’. Proc. Second IEEE Workshop Mobile Computing Systems and Applications, 1999, pp. 90–100

IET Wirel. Sens. Syst., 2012, Vol. 2, Iss. 4, pp. 369–376
doi: 10.1049/iet-wss.2012.0014