

Energy optimize design for Data Encrypted Wireless Sensor Node

Miss Nikita Jaideo Gotmare¹, Mr. Dinesh Rotke²

¹(Department of electronics/ R.T.M.N.U, India)

²(Department of electronics/ R.T.M.N.U, India)

Abstract: Dynamic Energy utilization stays as a significant obstruction for full organization and misuse of remote sensor system (WSN) engineering these days. This paper introduces the outline and execution of a vitality mindful sensor hub, which can help in building vitality proficient Wsns. A vitality proficient methodology, which goes for minimizing vitality utilization from both the sensor hub level and the system level in a WSN, is proposed. To minimize the correspondence vitality utilization of the sensor hub, the separation between the transmitter and the recipient is evaluated before accessible transmission, and after that, the most reduced transmission force required to transmit the estimation information is figured and decided. The sensor hubs are likewise situated to rest mode between two continuous estimations for vitality sparing in typical working conditions. Moreover, vitality sparing can be attained by evaluating the vitality utilization inside the entire system under distinctive system arrangements and afterward by picking the most vitality effective one.

Keywords: Energy effectiveness, intermittent slumber/wake-up plan, accepted indicator quality evidence (RSSI), remote sensor system (WSN)

I. Introduction

Quick DEVELOPMENT of framework scaling down, remote association, and on-chip sign transforming has pushed the advancement of remote sensor mastery, which has enriched its wide submissions from condition established upkeep to created framework observing and biological feeling. The amount of remote sensors, which are ordinarily prompted as a remote sensor network(wsn), made for genuine applications has immediately expanded in most recent years, and this pattern is required to much more help in the one years from now . despite the fact that, power use still proceeds as a significant obstruction for the full organization and abuse of this engineering, albeit electric batteries could be energized, e.g. through sun powered vitality reaping systems. Former investigates have explored diverse advances, for example, obligation cycling and information driven advances, for diminishing force consumption. The movement versatile middle of the road access convention has been intended to decline power utilization by allowing sensor hubs to assume a low-control dormant state when they are not working in transmission or getting mode. Information driven advances could be part up into two unique classes: information packing and vitality effective information obtaining. In an alternate study, the sensor lattice was separated into a few subsystems, and just abnormal amount derivations are telecast between the subsystems. Along these lines, the vitality use for correspondence diminishes as the raw numbers to be passed on decrease. For vitality effective statistical data points securing, a versatile examining calculation comprising of obligation cycling and versatile attempting is recommended to decrease power use in a sensor network. Examiners have additionally examined different methodologies for vitality mindful transmission, including adjustment climbing outlines, multihop steering plans, system seperating, and low-control fittings. Moreover, a mix of slumber masterminding with obstruct transmission approach has been proposed to accomplish force keeping in a remote sight and sound sensor system. In this paper occasional slumber/wake-up configuration alongside information encoding is included into the sensor hub considers to more remote achieve the hub level vitality keeping.

II. Overview Of The System

The system consists of four sensor nodes .First one is configured as co-ordinator,Second one is as a router third and fourth are the end devices as shown in the block diagram below.The co-ordinator will be connected to the CMU i.e. central monitoring unit and it will collect the data from another sensor nodes and will give that data to the CMU.The co-ordinator is also designed to send the data to other sensor nodes.The Second sensor node which is configured as a router transmits the data to the end devices and vice a versa.

III. Sensing Design

There are several components in multi-path routing protocol to construct multiple paths and distribute the traffic over the discovered paths. The performance gains of the multi-path routing protocols are highly dependent on the ability of the proposed protocol to construct high quality, reliable paths. We describe these components in details.

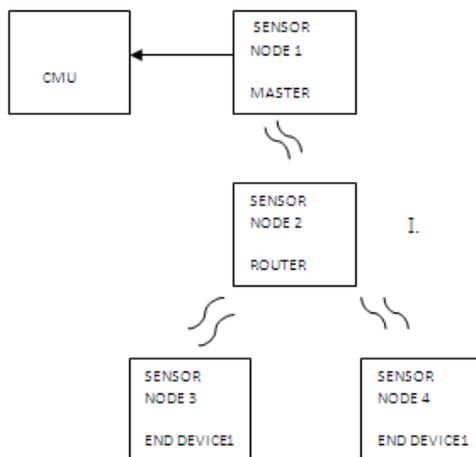


Fig.2. Block diagram of a system.

3.1.1 Path Discovery

As we know that data transmission in wireless sensor networks are done in multiple hops so choosing the intermediate nodes for creating multiple paths from source to the destination is one of the most important task. Among the aforementioned parameters path disjointness is the main criterion which is utilized by all the existing protocols. The path disjointness is categorized as node disjoint, link disjoint and partially disjoint. Node disjoint means there will be no common nodes in the discovered paths. Link disjoint paths may have several nodes in common, while there will be no shared link between the paths. Partially Disjoint paths include paths which share several links or nodes between paths.

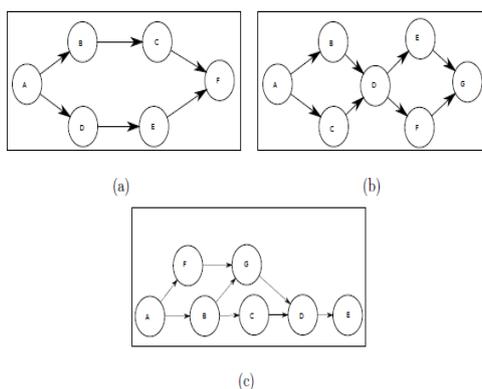


Fig.3.1.

3.1.2 Path Selection and Traffic Distribution

After the construction of paths there are many factors which can be used to select a path from multiple paths to transmit data from source to destination.

Some routing algorithms use the best path to transmit data and keep the others for backup, some may use the paths concurrently to transfer the data through multiple paths for reliability or even traffic distribution. The basic criteria considered for discovering a set of paths is the path disjointness but due to time varying properties of radio communication and resource limitation, considering only this factor can lead to construction of some low quality paths. To overcome this problem in addition to the amount of path disjointness various routing algorithms use various routing costs to make the best routing decision. Path length, packet loss rate, delay, residual battery level are some of the basic components of routing cost function. Once the set of paths is selected the routing protocol should determine how to distribute the network so that the resource utilization is maximized, improve performance demands such as throughput, data delivery ratio, delay, life time etc.

3.1.3 Path Maintenance

Due to the resource constraints of the sensor nodes and high dynamics of low-power links paths are highly error prone. Therefore there should be mechanism for path reconstruction to reduce performance degradation. The path discovery can start in three different situations,

1. When an active path has failed.

2. When all active paths have failed.
3. When certain number of paths have failed.

The proposed protocol named Energy Aware Multipath Routing in ZigBee based Wireless Sensor Networks is based on the multipath scheme where multiple routes exist between each source and the sink. In the following section we discuss system model and assumptions for the proposed protocol and working principle of the proposed protocol.

3.2 Network Model and System Assumption

There are n number of sensor nodes and a sink node in the network. After the deployment the nodes has to be stationary. We consider the sink node is in the middle of the network and static in nature. The sink node possesses unlimited computation, memory, and battery power. Each node knows their position, and the sink node contains the ID and location of the source node in the network. Sensor nodes are densely deployed and all are homogeneous. Each nodes communication range is identical and predefined.

3.3 Energy Aware Multipath Routing

We propose a routing algorithm which avoids flooding and takes the benefit of both load balancing and collision aware mechanism for energy conservation. Proactive routing protocol is preferred for the static network, but it is not advisable for the resource constrained sensor network, because in proactive protocols each node broadcasts messages to the entire network if there is a change in the network topology to keep the updated information and hence incurs an extra overhead. So we construct the route between source and sink when actually sink need the data from a particular source node. With these requirements we design a multipath routing algorithm for WSN. It mainly consists of three phases: Neighbor Discovery, Multipath Construction, and Data Transmission.

3.3.1 Neighbor Discovery

In this phase every node broadcasts a control packet contains their node ID, residual energy, and the location and wait for the neighbor discovery control packets from the nodes of its range to find the neighbor nodes. After the neighbor discovery phase each node finds its neighbor nodes. Now every node has the partial view of the network as illustrated in the Figure

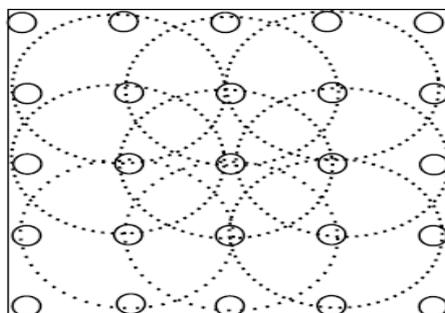


Fig.3.2.

3.3.2 Multipath Construction

After the Neighbor Discovery phase, each node possesses their neighbor information and then the Multipath Construction phase starts. We assume that the source node location is known to the sink and based on the location of the source the sink starts the route request process. In this the main concept is that, there are two types of nodes primary and alternate. A node is a primary node if it is in the primary path from source to sink else if it is the part of any alternate path then it is the alternate node. As described in the Algorithm 1, the primary nodes find two paths to the source, the primary path and the alternate path. The primary path is built with the best possible neighbor (having the minimum Location Factor(LF))and the alternate path is constructed with the next best neighbor (having the next minimum Location Factor(LF) after the primary path node).The alternate nodes find one single path towards the source node and searches its neighbor table for the node with minimum Location Factor(LF) and will prefer a primary node if possible, this is done to converge the path else the path can diverge from its direction toward the source, Next hop is chosen by the following equations

$$\text{Next-hop}_i = \min (L F_i) \text{-----}(3.1)$$

$$L F_i = \frac{(\text{Loc}_{\text{source}} - \text{Loc}_b)^2}{2 \text{Neighbor}_i} \text{-----}(3.2)$$

Where, $L F_i$ is the set of distance of all the neighbors of node i from the source. $\text{Loc}_{\text{source}}$ is the location of the source node, Loc_b is the location of the node b , and Neighbor_i is the neighbor set of node i .

Here it is an incremental approach from the sink to the source. First the sink node which is itself a primary node, selects two neighbors based on the equation 3.1. Out of these two neighbor nodes one with the minimum location factor becomes the next primary node and the node with the second minimum location factor becomes the alternate node, and with this step we initialize the multipath construction phase. As shown in figure, node a which is connected by bold line has the minimum location factor, signifies a primary node and is in the primary path towards the source. Similarly node b is connected by dashed lines has the second minimum location factor is the alternate node and is the part of the alternate path towards the source node. All the intermediate nodes follow the same process as the sink node to find their corresponding neighbors till the source node is reached, this process is illustrated in the figures 3.2(b), 3.2(c), 3.2(d) and 3.2(e). Finally when the route request reaches the source node we see that one primary path and multiple alternate paths are constructed between the sink and the source node which is shown in the figure

Input: Set of n sensor nodes randomly distributed.

Output: One primary and multiple alternate paths from source to sink.

```
repeat
if (node == sinknode) then
FindPrimaryPath();
FindAlternatePath();
else if (node == Primary) then
FindPrimaryPath();
FindAlternatePath();
else if (node == Alternate) then
FindPrimaryPath();
end if
until (node 6= Source)
procedure FindPrimaryPath()
if (node == Primary) then
Broadcast PRIMARY;
Search for the best node;
node ← Primary;
end if
if (node == Alternate) then
Broadcast ALTERNATE;
Search for the best node and prefer Primary;
if (node 6= Primary) then
node ← Alternate;
end if
end if
end procedure
procedure FindAlternatePath()
if node == primary then
Search for the next best path node accept Primary;
if ((node 6= Primary)&&(node 6= Alternate)) then
node ← Alternate;
end if
end if
if (node == Alternate) then
Exit();
end if
end procedure
```

Algorithm 1 has two procedures FindPrimaryPath() and FindAlternatePath() which are repeated till the route request reaches the source node. FindPrimaryPath() : This function is called by both primary and the alternate nodes. If the node is primary node it will broadcast its node type to be primary among its neighbors and search the node with minimum location factor in direction of the source node. Similarly if its an alternate node it broadcasts its node type to be alternate and finds the node with minimum location factor towards the source and will prefer the primary node if possible, so that the path converge instead of diverge. In both the above cases the found neighbor nodes can have two possible node types,

1. The node can be a primary node
2. Or it can be an alternate node

else it has not been assigned any node type. If the parent node is a primary node then the node type of the found neighbor in any of the above cases will be changed to primary node. In case the parent node is an alternate node, the node type of the found neighbor will not be changed if it has already been assigned a node type, and in case it has not been assigned any node type, the node will be assigned as alternate node. Find Alternate Path(): This function is called only by the Primary nodes for finding an alternate path towards the source. It finds the next best node which is called alternate node and adds it in its path. In the algorithm all nodes except the primary nodes are put to sleep mode. At a time there is only one active path between the source node and the sink node.

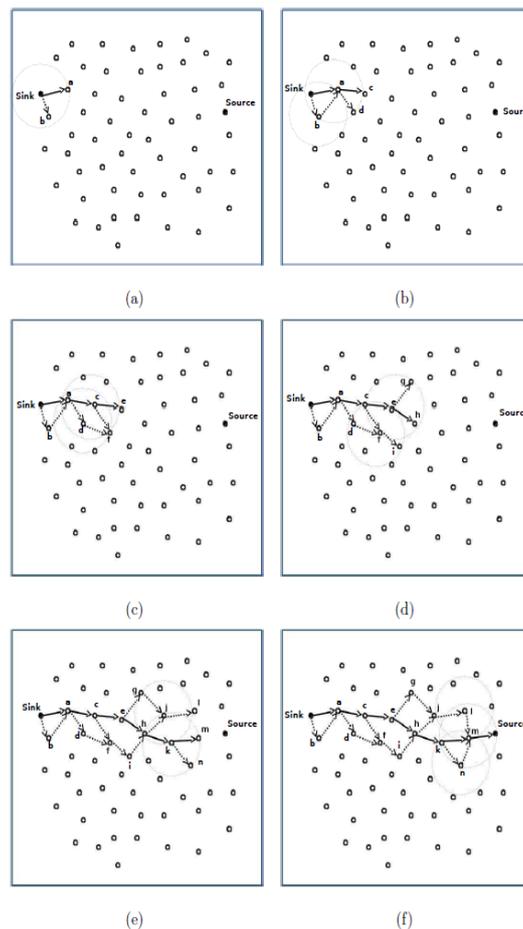


Fig.3.3.

This is done to reduce interference from other paths and avoid collision. Both of these factors help to save energy. If the primary path disrupts the protocol selects the alternate path with the best metric(e.g. hop count) to transmit data, and if all path disrupts and no path is left between the source and sink then again the process starts from the Neighbor Discovery phase.

3.3.3 Data Transmission

After the route discovery by the multipath construction phase data transmission takes place between source and sink. The primary and alternate paths are available, but the data transmitted only over the primary path. Source utilize alternate path when primary path is not available. Rest of the nodes that are not in the active path will go to sleep mode to conserve energy. If there will be no path exist between source and sink, the process of route discovery starts.

3.4 Energy Model

Energy modeling in WSN is based on the theoretical energy consumption of the existing platforms. We have considered three modes of energy consumption, first the energy consumed due to transmission of packets

(Eq 3.3), secondly, energy consumed due to reception of packets (Eq 3.4), third, energy spent by nodes in the idle mode (Eq 3.5) and finally the energy consumed by the nodes in processing.

$$\text{Energy Transmission} = \text{Energy}_{TX} \times t(\text{bits}) + \text{EXP}(d_2) \text{-----} (3.3)$$

$$\text{Energy Receiving} = \text{EXR} \times t(\text{bits}) \text{-----} (3.4)$$

$$\text{Energy Sleep} = \text{EXS} \times t(\text{sec}) \text{-----} (3.5)$$

$$\text{Total Energy} = \text{Energy Transmission} + \text{Energy Receiving} + \text{Energy Sleep} \text{-----} (3.6)$$

In Equations 3.3, 3.4 and 3.5 Energy XT refers to energy consumed per bit for transmission, EXR is the energy consumed per bit for receiving, and EXS is the energy consumed per second in idle mode and EXP (d2) is the energy consumed in finding the next hop neighbor.

A .Communication module

In this paper, Chipcon CC2500 is chosen as the transceiver used in the sensor node design. The CC2500 is a low cost true single chip 2.4GHz transceiver designed for very low power wireless applications. The received signal power PRx in CC2500 is translated to a decibel (dBm) value—RSSI. The relationship between PRx and RSSI is defined as follows:

$$\text{RSSI} = 10 \log_{10} \text{PRx} \text{-----} (3.7)$$

In Rx mode, the RSSI value can be read continuously from the RSSI status register, which is a binary complement number. The following procedure can be used to convert the RSSI reading to an absolute power level (RSSIdBm).

- 1) Read the RSSI status register.
- 2) Convert the reading from a hexadecimal number to a decimal number (RSSIdec).
- 3) If RSSIdec ≥ 128, then RSSIdBm = (RSSIdec – 256)/2 – RSSIoffset.
- 4) Else if RSSIdec < 128, then RSSIdBm = RSSIdec/2 – RSSIoffset

Where RSSIoffset is a typical value corresponding to the data rate and the center frequency. we can further obtain the following relationship as

$$10 \cdot \log_{10} P_{Tx} = 10 \cdot \log_{10} P_s + 10 \cdot \log_{10} P_{Tx \max} - 10 \cdot \log_{10} P_{Rx} \text{-----} (3.8)$$

Where 10.log10 PTx is the transmission power in dBm and 10.log10 PRx is the RSSI. There is a register PATABLE in CC2500 which is used in selecting the transmission power. From (3.8), we can compute the minimum required power as

$$\text{PATABLE} = 10 \log_{10} P_s + 10 \cdot \log_{10} P_{Tx \max} - \text{RSSI}$$

The format of the data packet can be configured and consists of the following items (see Figure 3.4):

- Preamble
- Synchronization word
- Length byte or constant programmable packet length
- Optional address byte
- Payload
- Optional 2 byte CRC

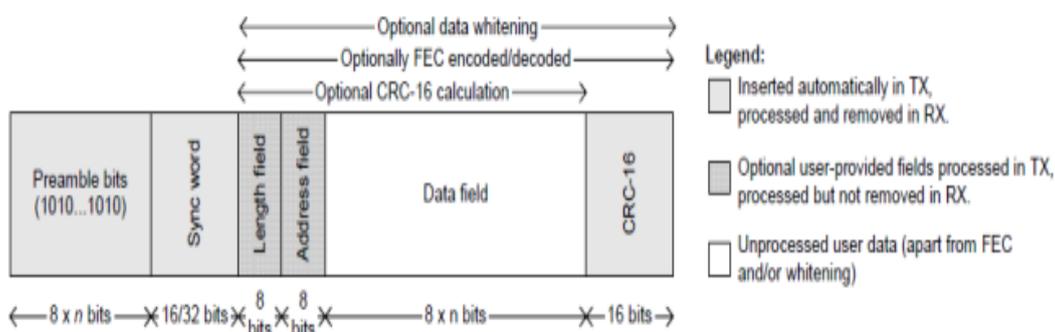


Fig. 3.4. Data packet in this paper.

Each item can be configured by setting the corresponding registers when initializing CC2500. In this paper, the data packet is configured as shown in Fig. 3.4. As a result, each sensor node will transmit (10 + n) byte data one time and the energy consumption can be computed as follows:

$$E_{Tx} = P_{Tx} \cdot \frac{(10 + n) \cdot 8}{R}$$

Where n denotes the number of payload data bits needed to be transmitted. Here, temperature data are the payload data and $R = 100$ kb/s is used in this paper.

B. Occasional Sleep/Wake-Up Scheme

Aside from the versatile RF force give as related in the first part, the hub level vitality keeping is moreover fulfilled through the intermittent slumber/wake-up configuration. As we know whether a WSN is created in remote ranges or under unpleasant situations where physically reviving electric batteries for sensors is not plausible, one commonplace substitute methodology for force keeping is to turn off a few sensors and to initiate just an important set of sensors while giving a decent feeling medication and lattice network at the same time. In progressively changing parameter estimations, for example, warmth, not all sensor hubs are obliged to stay in dedicated mode. In this manner, in arrangement to spare force use, the sensor hubs are imagined to be put into a nap mode with a clock that works out their rest length. At the point when the clock floods, a cut off happens, and it will wake those hubs up and will then present estimations and raw numbers transmission.

C. Sensor Node Design

Every sensor hub ought to have the capacity to gather natural parameters and correspond with one another. In the meantime, its comparing equipment and programming ought to be vitality effective and incorporate the usefulness depicted in past areas. In view of these necessities, the sensor hub has been composed and actualized as indicated in Fig. 4. In this outline, the Pic18f252 microcontroller is picked as the center for satisfying reckoning and control capacities of the sensor hub; this is on account of it has the competence of low power utilization and short waking-up .

D. System Level Energy-Saving Realization

In spite of the fact that the vitality utilization of every sensor hub is exclusively minimized by its partner utilitarian modules, the aggregate vitality utilization might be further decreased by utilizing the proper sensing plan for the entire sensor network. First, the sensor hub is situated to rest mode until the clock floods, and afterward, it is waked up to gather ecological parameters and holds up to speak with the CMU. Second, the proposed plan at first gauges the base obliged transmission power P_{tx} by utilizing the test code from the CMU or the group head. At that point, the CMU chooses a fitting sensing plan by thinking about the aggregate vitality utilization. By accepting a homogeneous equipment plan for the greater part of the sensor hubs, the variables speaking to separations between sensor hubs/bunch heads and CMU are amassed by television appeal to the majority of the sensor hubs. After all data is gathered, the CMU gauges general vitality utilization for each of the plans and settles on choice to pick the unified with the best vitality effectiveness as the current system plan. In this plan, test code is characterized as the summon for sensor hubs to execute diverse undertakings. Here, test code 1 educates the greater part of the sensor hubs to gauge patable1, test code 2 advises the majority of the group heads to show test code 3 in the neighborhood bunch, test code 3 illuminates the sensor hubs in the nearby group to gauge patable2, test code 4 advises the greater part of the hubs to select patable1 as its transmission power and to set the CMU ID as its goal address, and test code 5 illuminates the greater part of the hubs to select patable2 as its transmission power and to set its group head ID as its end address.

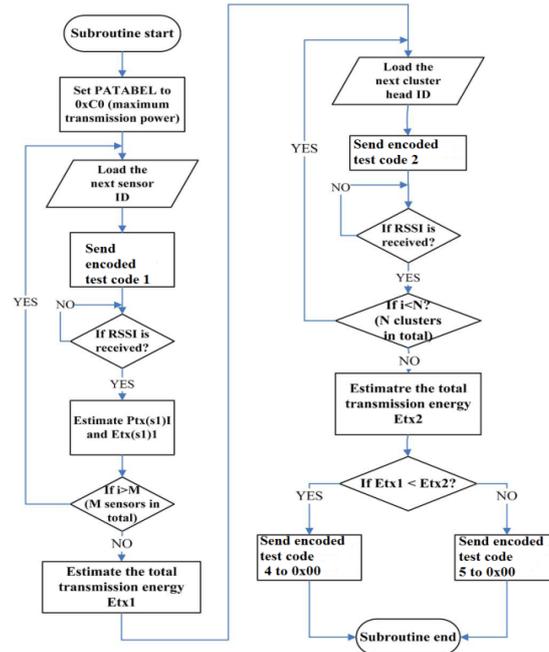


Fig. 3.5. Flowchart for energy estimation and scheme selection on the CMU.

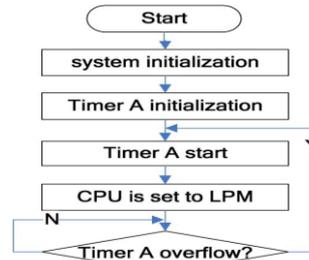


Fig. 3.6. Flowchart of the main program for timer initialization and low-power mode setting.

Fig. 3.5. Shows the flowcharts for energy-related information gathering, potential energy consumption calculation, and sensing scheme switching. The corresponding proposed algorithm is given in Algorithm 1.

Algorithm 1: CMU_Estimate(M,N)

1. PATABEL = 0xC0; //maximum transmission power
2. For (i = 0; i < M; i++){
3. Load the address of destination sensor node node_ID[i];
4. Send encoded test code1;
5. While (ReceivedF lag == 0);
6. $E_{Tx}(s1)[i] = 10^{10 \cdot \log_{10} P_S + 10 \cdot \log_{10} P_{Txmax} - RSSI_i / 10}$;
7. }
8. $E_{Tx}(s1) = \sum_{i=1}^M E_{Tx}(s1)[i]$;
9. For (j = 0; j < N; j++){
10. Load the address of destination cluster head ClusterHead_ID[j];
11. Send encoded test Code2;
12. While (ReceivedF lag == 0);
13. $E_{Tx}(s2)[j] = \sum_{i=0}^{H_j} 10^{10 \cdot \log_{10} P_S + 10 \cdot \log_{10} P_{Txmax} - \frac{RSSI_i \cdot ((10+n) \cdot \frac{8}{R})}{10}}$;
14. }
15. $E_{Tx}(s2) = \sum_{j=1}^N E_{Tx}(s2)[j]$;
16. If (ETx(S1) < ETx(S2))
17. Broadcast encoded test code4;
18. Else
19. Broadcast encoded test code5;

In Algorithm 1, M and N denote the number of sensor nodes and clusters in the network, respectively. The array Node_ID[i] and ClusterHead_ID[j] are defined as buffers to save the address of the sensor nodes and cluster heads, respectively. ReceivedF lag is defined as a flag to indicate the receiving state (ReceivedF lag = 0 indicates that the sensor node does not receive any data bits, and ReceivedF lag = 1 indicates that the sensor node receives all data.) Fig.3.6 shows the main program of the sensor node and the cluster head for timer initialization and sleep mode setting.

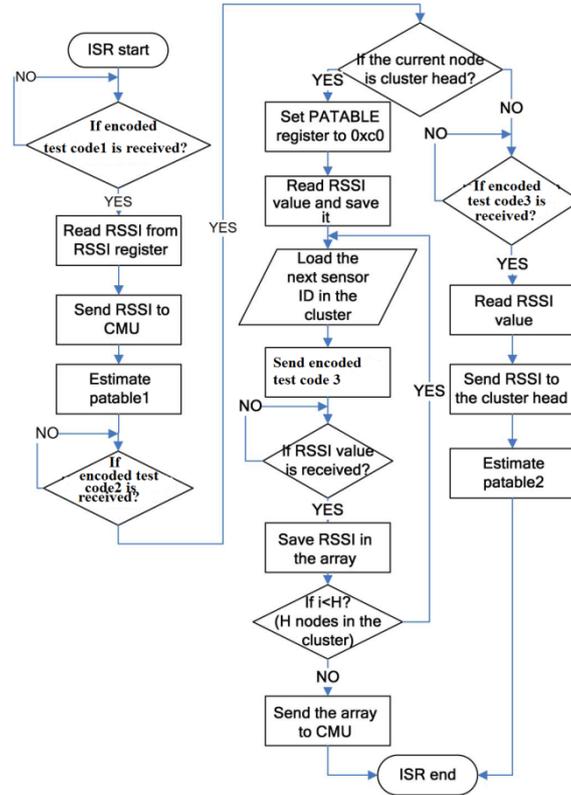


Fig. 3.7. Flowchart for energy estimation and scheme selection steps on sensors and cluster heads.

Fig. 3.7. Shows the flowchart of the interrupt service routine of the sensor node and the cluster head for estimating the minimum required transmission power and the scheme switching. The corresponding proposed algorithm is given in Algorithm 2.

Algorithm 2: Sensor_Estimate_ISR()

1. While(ReceivedF lag == 0);
2. If(Rx_encodedtestcode==encodedtestcode1){
3. Send RSSI to CMU;
4. $PATABLE1 = 10 \cdot \log_{10} P_S + 10 \cdot \log_{10} P_{Txmax} - RSSI$;
5. }
6. Else if(Rx_encodedtestcode == encodedtestcode 2){
7. If(ClusterFlag == 1){
8. Save the RSSI value of cluster head in the RSSI array RSSI [H];
9. For(i = 1; i < H; i++){
10. Save the RSSI value of cluster head in the RSSI array RSSI [H];
11. PATABLE1 = 0× c0; //maximum transmission power
12. Load the address of destination cluster node ClusterNodeID[i];
13. Send encodedtestcode 3;
14. While(ReceivedF lag == 0);
15. Save the RSSI value of the ith cluster head in the RSSI array RSSI[H];
16. }
17. Send the RSSI array RSSI[H] to CMU;
18. }
19. Else if(ClusterFlag == 2){
20. While(ReceivedF lag == 0);

21. Send RSSI to the corresponding cluster head;
22. $P_{ATABLE2} = 10 \cdot \log_{10} PS + 10 \cdot \log_{10} P_{Txmax} - RSSI$;
23. }
24. }

In Algorithm 2, ClusterFlag is a status flag used to indicate whether the sensor node is cluster head. ClusterNode_ID[i] is defined as a buffer to save address of the sensor nodes in clusters. RSSI[H] is used as a buffer to save the RSSI values of the sensor nodes in the clusters.

IV. Experimental Test

In this paper, the energy-efficient strategy that we designed on the sensor platform consists of two parts: node-level energy saving and network-level energy saving. The node-level energy saving is achieved by adaptive transmission power setting and by the periodic sleep/wake-up scheme. The network-level energy saving is achieved by adaptive network configuration. In the experimental test, we first investigate the performance of node-level energy saving. Then, the performance of network level energy saving is investigated by comparing the energy consumption of fixed network configuration with that of adaptive network configuration.

1. Calculated Results:

Energy level for individual node for 8 bit,16bit,32bit,64 bit is calculated first for both Huffman encoded as well as without Huffman encoded data then the graph of sensor node versus energy level is plotted for 8bit,16 bit,32bit & 64 bit data which is as shown in fig4.1(a), fig4.1(b), fig4.1 (c) & fig4.1 (d) respectively. Here WH denotes without Huffman encoding & HU denotes with Huffman encoding .By observing all four graphs it is seen that energy level for Huffman encoded data is always greater than without Huffman encoded data whether it is 8B,16B,32B or 64B data.

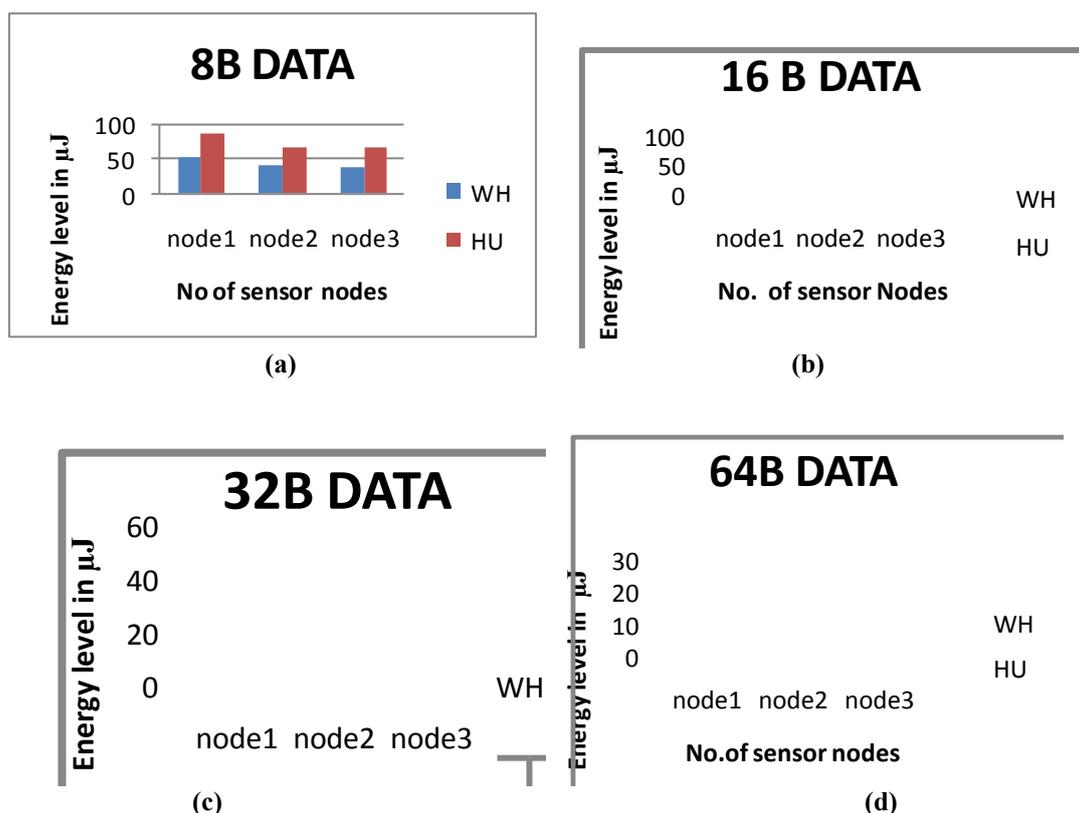


Fig.4.1(a),(b),(c),(d) Graph of sensor node versus energy level for 8bit,16bit,32bit,64bit data respectively Fig 4.1(e), Fig 4.1 (f) & Fig 4.1(g) given below shows the graph of data packet length versus energy level for each individual sensor node. Here also it is seen that energy level for Huffman encoded data is always greater than that of without Huffman encoded data.

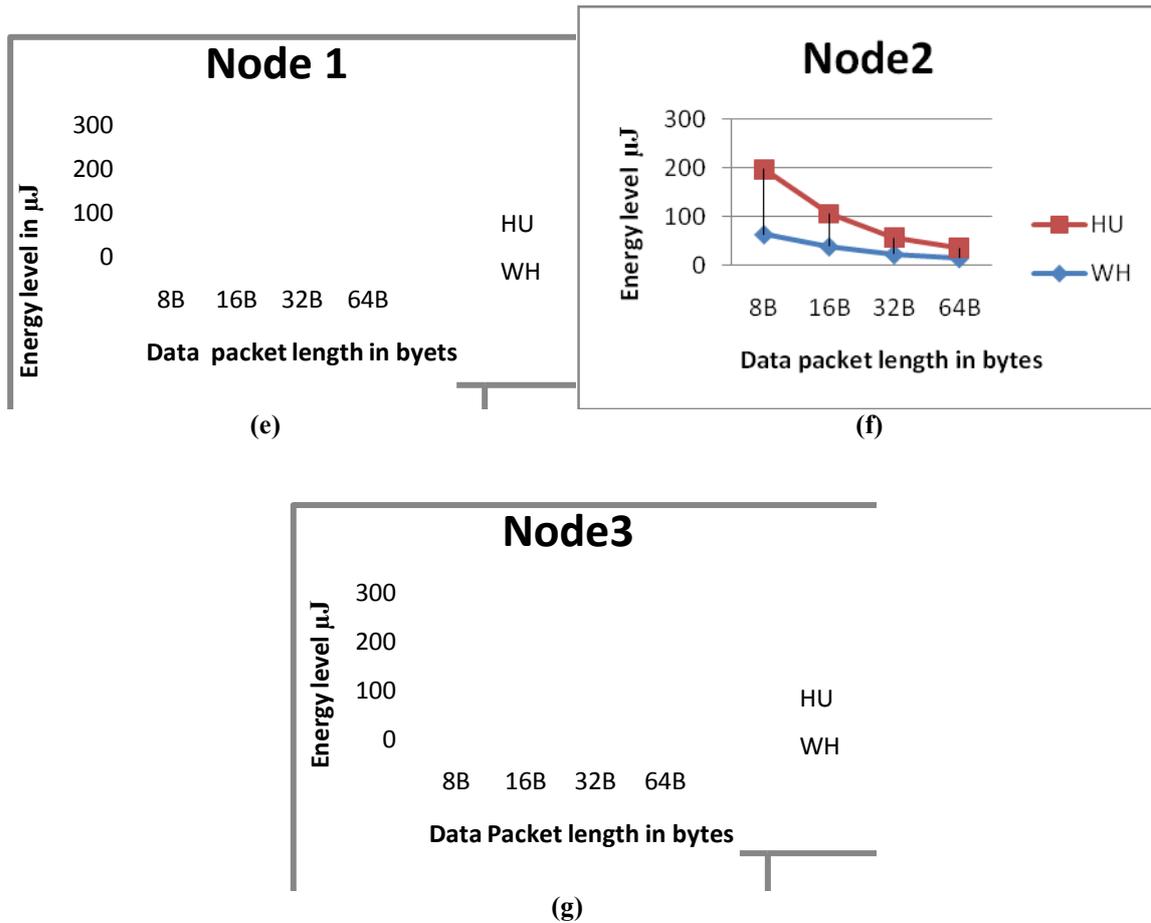
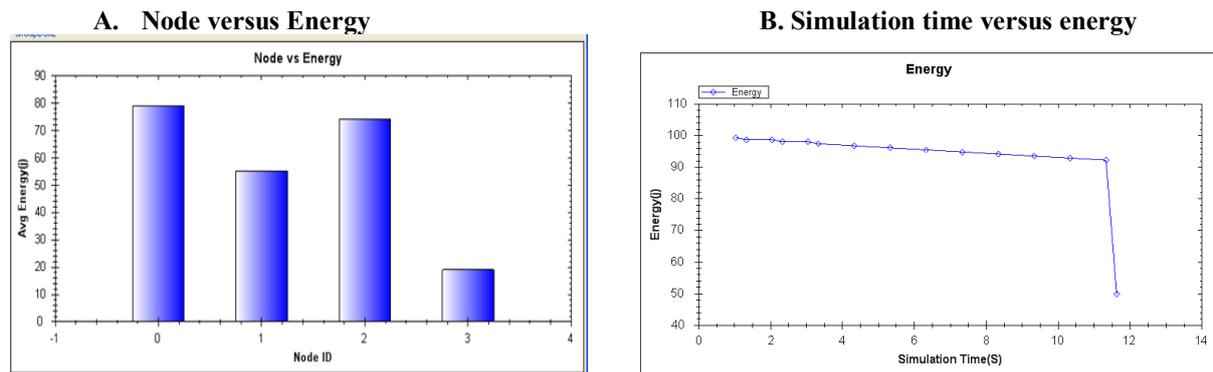


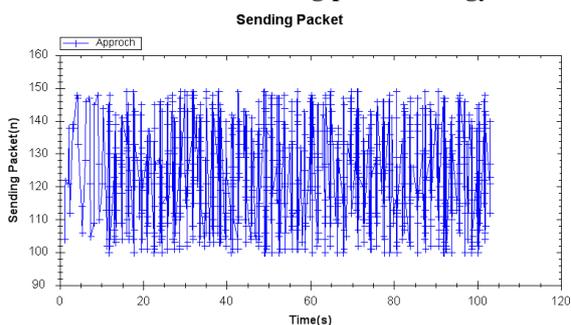
Fig.4.1 (e), (f) & (g) Graph of data packet length versus energy level for sensor node 1,2 & 3 respectively

2. Results on software:

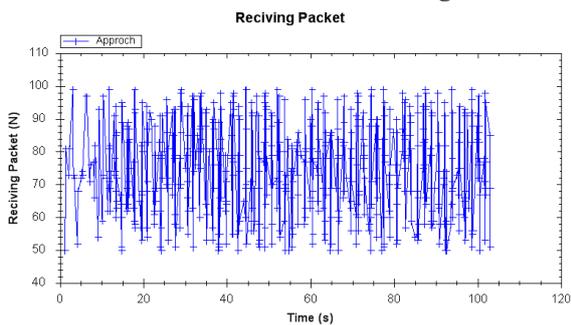
Graphs below are taken from software simulation results which done on VB.net. The simulation can be done for two modes, one is without Huffman encoding mode & another is with Huffman encoding mode and also the results are obtained for both modes.



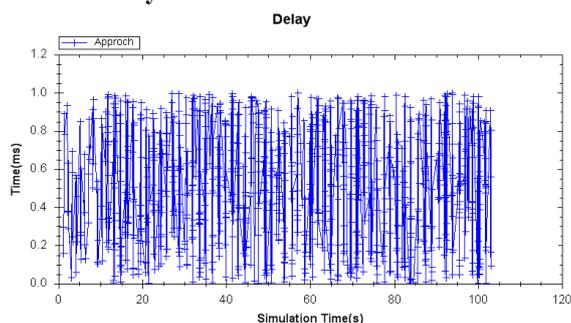
B. Time versus sending packet energy



D. Time versus Receiving Packet



E. Delay



Data packet	Energy Mean(μ J)		Energy level (% increase)
	WH	HU	
8B	69.67	145.33	108.59
16B	42.33	72.67	71.67
32B	25.33	37	46.07
64B	16.67	23	27.52

Table (a): Statistical Analysis of total energy consumption

Table (a) given below shows the percentage increase energy level for due to Huffman encoding of data. As seen in the table below WH denotes mean energy level for without Huffman encoded data & HU denotes the mean energy level with Huffman encoding of data which is always greater as compared to without Huffman encoded data. Last column shows the percentage increase in energy level.

Data packet		Without Huffman encoding(WH)		With Huffman encoding(HU)		Percentage Compression(WH)= (S - R)/S * 100	Percentage Compression(HU)= (S - R)/S * 100
		Sending Packet(S)	Receiving Packet(R)	Sending Packet(S)	Receiving Packet(R)		
8B	N1	53	50	25	21	5.66	16
	N2	64	60	30	21	6.25	30
	N3	65	61	32	21	6.55	34.37
16B	N1	87	81	51	36	6.89	29.41
	N2	106	99	61	36	6.66	40.98
	N3	108	101	63	36	6.48	42.85
32B	N1	145	135	100	67	6.89	33
	N2	174	163	122	67	6.32	45.08
	N3	180	168	125	67	6.67	46.4
64B	N1	216	202	158	96	6.48	39.24
	N2	260	243	190	96	6.53	49.47
	N3	267	249	196	96	6.74	51.02

Table(b): Analysis of percentage compression

Table (b) given below shows the packet delivery ratio which is given by $P=R/S$;

Where P is the packet delivery ratio, R is receiving packet, S is sending packet. Also it gives percentage compression without Huffman encoding & with Huffman encoding, which is given as
Percentage compression= $(S-R)/S*100$

And as we can see from the calculated results as shown below in table (b) percentage compression in Huffman encoded data is much more greater than that of without Huffman encoded data. And again it increases with data length.



Fig4.2(a) Sensor node designed which is configured as coordinator

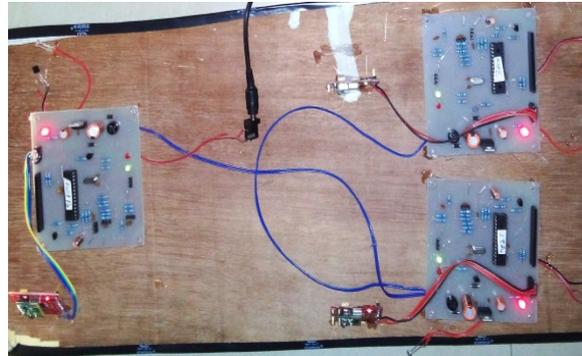


Fig4.2 (b) Sensor nodes developed for temp., light intensity & voltage monitoring

V. Conclusion

In this paper, we proposed an energy efficient multipath routing protocol for WSN. This protocol is designed to decrease the routing overhead, improve the latency and packet delivery ratio and through discovering multiple paths from the source to the destination. It has a sink initiated Route Discovery process with the location information of the source known to the sink. There are two types of nodes which are used here one is primary and the other is alternate. At the end of the route formation one primary path and multiple alternate paths are built and all nodes except the primary paths nodes are put to sleep mode which helps us to save energy and generate a collision free environment, the primary path is used to transmit the data from source to the sink and if the route disrupts, the next best alternate route is used for the purpose and if no path exists between the source and destination then the route discovery algorithm calls. The simulation result finds the latency, packet delivery ratio, average control packet over head and total energy consumed.

Acknowledgement

This paper presents a part of the research performed in the project: “Energy optimize design for wireless sensor node using data encoding”. The authors are grateful for their support.

References

- [1]. C. Chong and S. P. Kumar, “Sensor networks: Evolution, opportunities, challenges,” Proc. IEEE, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
- [2]. R. Gao and Z. Fan, “Architectural design of a sensory node controller for optimized energy utilization in sensor networks,” IEEE Trans. Instrum.Meas., vol. 55, no. 2, pp. 415–428, Apr. 2006.
- [3]. A. Tiwari, P. Ballal, and F. L. Levis, “Energy-efficient wireless sensor network design and implementation for condition-based maintenance,” ACM Trans. Sens. Netw., vol. 3, no. 1, pp. 1–23, Mar. 2007.
- [4]. F. Salvadori, M. Campos, P. Sausen, R. Camargo, C. Gehrke, C. Rech, M. Spohn, and A. Oliveira, “Monitoring in industrial systems using wireless sensor network with dynamic power management,” IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3104–3111, Sep. 2009.
- [5]. D. Gallo, C. Landi, and N. Pasquino, “Multisensor network for urban electromagnetic field monitoring,” IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3315–3322, Sep. 2009.
- [6]. ON World Inc., Wireless Sensor Networks: Growing Markets, Accelerating Demands, Jul. 2005. [Online]. Available: <http://www.onworld.com/html/wirelessensorsrprt2.htm>
- [7]. Embedded WiSeNTs Consortium, Embedded WiSeNTs Research Roadmap (Deliverable 3.3). [Online]. Available: <http://www.embedded-wisents.org/>
- [8]. C. Alippi and C. Galperti, “An adaptive system for optimal solar energy harvesting in wireless sensor network nodes,” IEEE Trans. Circuits Syst.I, Reg. Papers, vol. 55, no. 6, pp. 1742–1750, Jul. 2008.
- [9]. A. Giuseppe, C. Marco, and D. F. Mario, “Energy conservation in wireless sensor networks: A survey,” Ad Hoc Netw., vol. 7, no. 3, pp. 537–568, May 2009.
- [10]. C. Schurgers, V. Tsiatsis, and M. B. Srivastava, “STEM: Topology management for energy efficient sensor networks,” in Proc. IEEE Aerosp. Conf., 2002, pp. 3-1099–3-1108.
- [11]. V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy efficient, collision-free medium access control for wireless sensor networks,” Wireless Netw., vol. 12, no. 1, pp. 63–78, Feb. 2006.
- [12]. A. Kadrolkar, R. Gao, and R. Yan, “Energy efficient data transmission for manufacturing system health monitoring,” in Proc. 9th Int. Conf. Frontiers Des. Manuf., Changsha, China, Jul. 2010, pp. 65–70.

- [13]. R. Yan, D. Ball, A. Deshmukh, and R. Gao, "A Bayesian network approach to energy-aware distributed sensing," in Proc. IEEE Sens. Conf., Vienna, Austria, Oct. 2004, pp. 44–47.
- [14]. C. Alippi, G. Anastasi, M. D. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," IEEE Trans. Instrum. Meas., vol. 59, no. 2, pp. 335–344, Feb. 2010.
- [15]. V. Raghunathan, C. Schurgers, and S. Park, "Energy-aware wireless microsensor networks," IEEE Signal Process. Mag., vol. 19, no. 2, pp. 40–50, Mar. 2002.
- [16]. C. Schurgers, V. Raghunathan, and M. Srivastava, "Power management for energy-aware communication systems," ACM Trans. Embedded Comput. Syst., vol. 2, no. 3, pp. 431–447, Aug. 2003.
- [17]. E. Biyikoglu, B. Prabhakar, and A. Gamal, "Energy efficient packet transmission over a wireless link," IEEE/ACM Trans. Netw., vol. 10, no. 4, pp. 487–499, Aug. 2002.
- [18]. R. Gao, A. Deshmukh, R. Yan, and Z. Fan, "Energy efficient wireless sensor network for dynamic system monitoring," in Proc. SPIE, 2005, vol. 5999, no. 4, pp. 1–10.
- [19]. D. Ball, R. Yan, T. Licht, A. Deshmukh, and R. Gao, "A strategy for decomposing large-scale energy-constrained sensor networks for system monitoring," Product. Plann. Control, vol. 19, no. 4, pp. 435–447, 2008.
- [20]. A. Wang, W. Heinzelman, A. Sinha, and A. Chandrakasan, "Energy scalable protocols for battery-operated micro sensor networks," J. VLSI Signal Process. Syst. Signal Image Video Technol., vol. 29, no. 3, pp. 223–237, Nov. 2001.
- [21]. C. Sha, R. Wang, H. Huang, and L. Sun, "An energy-saving strategy based on sleep scheduling and block transmission for wireless multimedia sensor networks," Int. J. Pervas. Comput. Commun., vol. 6, no. 2, pp. 248–267, 2010.
- [22]. H. Sun, Y. Qian, and R. Yan, "Design and realization of an intelligent sensor node with its application in energy-aware WSNs," in Proc. IEEE I2MTC, Graz, Austria, May 11–13, 2012, pp. 941–946.
- [23]. A. Wang and A. Chandrakasan, "Energy-efficient DSPs for wireless sensor networks," IEEE Signal Process. Mag., vol. 19, no. 4, pp. 68–78, Jul. 2002.
- [24]. A. Chehri, P. Fortier, and M. Tardif, "UWB-based sensor networks for localization in mining environments," Ad Hoc Netw., vol. 7, no. 5, pp. 987–1000, Jul. 2009.