ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

Link Quality Aware Wireless Sensor Networks Using Adaptive Code Dissemination Protocol

C. Bhuvaneswari[1], Ellcy Priana[2]

Student, Dept. of Computer Science, Sir Issac Newton College of Engineering and Technology,

Anna University, Pappakovil, Nagapttinam, India[1].

Assistant Professor in Department of Computer Science, Sir Issac Newton College of Engineering and Technology,

Anna University, Pappakovil, Nagapattinam, India[2].

ABSTRACT: Wireless reprogramming is a crucial technique for software deployment in wireless sensor networks. Code Dissemination is basic building block to enable wireless reprogramming. We present a Adaptive code Dissemination protocol. It is used to solve hidden terminal problems. It uses fast sender selection algorithm to avoid transmission collision. We enable dynamically configurable packet sizes to support large packets to improve efficiency. It employs a simple impact-based back off timer design to shorten the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit.

KEYWORDS: wireless sensor networks; dissemination; reprogramming

I. Introduction

Wireless Sensor Networks applications often need to be changed after deployment for a variety of reasons. Reconfiguring a set of parameters, modifying tasks of individual nodes, and patching security holes. Many large scale WSN, are deployed in environments where physically collecting previously deployed nodes is either very difficult or infeasible. A wireless sensor networks (WSN) of spatially distributed autonomous sensors top monitor physical or environmental conditions such a temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modem networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring and so on. Provide a bridge between the real physical and virtual worlds. Allow the ability to observe the previously observable at a fine resolution over large spatio-temporal scales. Have a wide range of potential applications to industry, science, transportation, civil infrastructure, and security.

Wireless reprogramming is a crucial technique for software deployment in wireless sensor networks. Code dissemination is a basic building block to enable wireless reprogramming. In a sensor networks when we transmit a packets from sender to receiver, packet will be transmitted from more than one node. If a transmitting node fails the packet will be lost. Sender did not know whether the packet is reached or not. Acknowledgement will not be received. Data will be lost and it takes more time to complete the process. To rectify this problem, is a node fails it is denoted to sender and so the sender transmit a packet again.

Existing code dissemination protocols Deluge and MNP adopt several key techniques to ensure high reliability and performance. First, they exchange control-plane message for high reliability [4][5]. Second, they segment a large code object into fixed size pages for pipelining [5]. The page transmission time and inter-page negotiation time(which involves exchanges of control-plane messages) are therefore two major contributors to the overall completion time. However, existing protocol designs exhibit their efficiency in two main aspects. First, the data throughput efficiency the ratio between the network throughput and PHY data rate degrades rapidly as the PHY rate increases. For example, given the packet size of approximately 36 bytes in both Deluge and MNP (both were originally designed for the 19.2 Kbps CC1000 radio), the efficiency ratio for the current 250 Kbps CC2420 radio is only 14.3 percent. Second, the current sender selection algorithm in MNP (for addressing the broadcast storm problem) does not consider link quality information and needs multiple rounds of message exchanges, resulting in transmitting redundancy and long completion time.

To address the first issue, we would like to increase the packet size to improve the transmission efficiency for high PHY rate radios (e.g., 250Kbps CC2420). This approach is appropriate for code dissemination because the traffic is always saturated and there are no delay constraints on individual packets. However, it would be inflexible to fix the packet size to its maximum allowable size as a fixed packet size may not be appropriate for all platforms under all the conditions [3]. Therefore, we support dynamically configurable packet sizes in our protocol design. To address the second issue, we leverage 1-hop neighbours' link quality information learned over the air to improve the sender selection accurate. We dynamically estimate the impacts of senders by considering both uncovered neighbours (i.e., neighbours that do not receive an entire page) and the link qualities to those neighbours. A node's transmission is considered more effective if the nodes have more uncovered neighbours with good link qualities. Considering link qualities help to put less weight an potential senders with poor link qualities to their neighbours, thus mitigating transmissions for accommodating low PRR(PRR is short for Packet Reception Ratio) receivers. This is especially important large packets are transmitted over the air. Given many candidate senders, our design needs to ensure that the best sender transmits while avoiding simultaneous retransmission attempts that can lead to duplicates pr collisions. MNP [5] needs multiple rounds of message exchanges and explicit requests from receivers, transmission overheads and long delays. We address this issue by proposing a fast sender selection mechanism that does not require explicit coordination. The basic ideas are to prioritize sender transmission so that the best sender with the largest impact is most likely to transmit.

We incorporate the above design principles into a new code dissemination protocol, ACD. ACD's performance extensively through a 25-node tested as well as TOSSIM simulations. Result show that

1. By supporting large packets, ACD significantly short ends the completion time for the TelosB platform with 250Kbps CC2420 radio.
2. By supporting accurate sender selection that leverages information learned over the air, ACD effectively reduces contentions and collisions, resulting a fewer packet transmission.
3. By the impact based backoff time design, ACD reduces the inter page negotiation time, also shortening the competition time.

Adaptive code dissemination protocol (ACD) is implement it based on the TinyOS operating system. ACD has four features. 1. It supports dynamically configurable packet sizes. By increasing the packet sizes for high PHY rate radios, it significantly improves the transmission efficiency. 2. Its employs an accurate sender selection algorithm to mitigate transmission collisions and transmission over poor links. 3. It employs a simple impact-based back off timer design to chart the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit. 4. It is used to avoid hidden terminal problems.

The basic principle of sender selection algorithm is to select the best sender for forwarding the data while avoiding simultaneous transmission from other neighbouring nodes. This involves two main aspects. First, an accurate metric should be devised to estimate sender's impacts. Second, efficient mechanisms should be designed to coordinate transmission of eligible senders so that the largest impact sender is most likely to transmit. Sender selection in broadcast protocols is a special case of sender selection in code dissemination protocols (i.e., one page, one packet). Sender selection in code dissemination protocols is more complex.

II. Related work

CORD [1] a reliable bulk data dissemination protocol for propagating a large data object to all the nodes in a large scale sensor networks. CORD's primary goal is to minimize energy consumption. To achieve its goals CORD employs a two phase approach in which the object is delivered to a subset of nodes in the network that forma connected dominating a set in the first phase and to the remaining nodes in the second phase. Further, CORD installs a coordinated sleep schedules on the nodes in the network whereby nodes that are not involved in receiving or transmitting data can turn off their radios to reduce their energy consumption. We evaluated the performance of CORD experimentally on both an indoor and outdoor sensor network test bed and via extensive simulations. CORD significantly reduces the energy consumption for reliable data dissemination while achieving a comparable latency.

Deluge [4] is perhaps the most popular code dissemination protocol used for reliable code updates. It uses a three-way handshake and NACK-based protocol for reliability, and employs segmentation (into pages) and pipelining for spatial multiplexing. Here we describe it in three different phases: 1. Advertisement (ADV), Request (REQ) and Page Transmission (DATA). In this phase, each node advertises about its local code objects. Note that the code object may not be complete during the process of transmission. Deluge enforces strict ordering of page transmission. When a node (receiver) learns that another node (sender) has more available pages, it will send a request and prepare to receive the data packets. When the sender receives a request, it will transition from the IDLE state to the TX state. Then it starts transmitting the requested data packets in the current page. 2. Page Retransmission. If a receiver loses some packets in a given page, it will remain in the RX state, sending requests to the most recently heard neighbour for the missing packets. On the other hand, the sender will enter the IDLE state whenever it completes transmitting a requested page. When it receives a request for transmission purpose, it sends the packets immediately. When all missing packets in a given page are received, the receiver enters the IDLE state and prepares to receive the next page. 3. New page Transmission. The sender and receiver will enter the IDLE state whenever a page completes. At this time, the receiver will prepare to receive the next page and increase the advertising frequency by resetting its advertisement timer.

When the sender hears an inconsistent advertisement from any of the receivers, it will also reset its advertisement timer to let the receivers learn about the fact that a neighbour with more available pages is in the vicinity. The receiver will request to the sender for the next page. After the sender receives a new page request, it will start transmitting immediately. A problem in Deluge is that when a sender receives request from receivers, it will start transmitting data packets after a specified timeout. It is probable that multiple senders in a neighbourhood start transmitting concurrently, causing serious collisions. To address this issue, MNP [5] incorporates a sender selection algorithm which attempts to guarantee that in a neighbourhood there is at most one source transmitting at a time. In MNP, source codes compete with each other based on the number of distinct request they have received.

The sender selection is greedy in that it tries to select the sender that is expected to have the most impact. MNP also reduces that active radio time of a sensor node by putting the node into “sleep” state when its neighbours are transmitting. This effectively reduces the idle listening problem and avoids overhearing. Compared to the above two works, our work has two main differences. First, we enable dynamically configurable packet sizes to support large packets to improve the dissemination performance. Second, we employ an accurate and fast sender selection algorithm to alleviate concurrent transmission and transmission over poor links.

Ad hoc network is a collection of wireless nodes dynamically forming a topology without any existing infrastructure. Because of the absence of locality, it is highly open to number of hidden and exposed terminal problems. Hence it is essential to design a MAC protocol with more consideration. A mobile agent based MAC [2] is introduced which roaming across the network, gathering information about dynamically changing environments. The main purpose of this paper is to ease the hidden and exposed terminal problems. ACKs and exploits link correlation to estimate the expected number of uncovered nodes while ACD relies on NAKs (i.e., REQ messages) and the link qualities to estimate the excepted number of uncovered receivers. Third, both protocols rely on back-off timers to prioritize the transmission. In a CF the back-off period is simply calculated as the reciprocal of the impact. In the ECD, the back-off period is more carefully optimized by minimizing the probabilities of “priority inversion” and transmission collisions.

Link estimation is important for sender selection. LEEP [7] is a passive link estimation protocol that can be invoked in proactive protocol to update neighbour’s link qualities. For example, LEEP can be built on top of a broadcasting service. The LEEP protocol attaches a LEEP header (including a packet sequential number) to the broadcast beacon. When a node receives a beacon, it can estimate the inbound link quality (i.e., the link quality from the current node to its neighbour) by inferring the number of missing packets. LEEP also attaches a LEEP footer (including inbound link qualities) to the broadcast beacon, so that each node can obtain the outbound link quality (i.e., the link quality from a neighbour to the current node) from the received beacon.

CF [6] is a recent work that exploits spatial link correlation to mitigate ACK overhead. ACD differs CF in three aspects. First, CF is used for flooding a single packet while ACD is used for disseminating large code objects consisting of multiple pages and packets. CF does not guarantee 100 percent reliability while ACD employs handshake and negotiation to achieve 100 percent reliability. Second, the basic intentions for sender selection are the same for both protocols, i.e., selecting the sender which can cover the most number of uncovered (i.e., not yet received) receivers in a neighbourhood. The techniques employed in those two protocols, however, are different. CF relies on ACKs and exploits link correlation to estimate the expected number of uncovered nodes while ACD relies on NAKs (i.e., REQ messages) and the link qualities to estimate the expected number of uncovered receivers. Third, both protocols rely on back-off timers to prioritize the transmissions. In CF the back-off period is simply calculated as the reciprocal of the impact. In ACD, the back-off period is more carefully optimized by minimizing the probabilities of “priority inversion” and transmission collisions.

III. PROPOSED ALGORITHM

A. *Adaptive Code Dissemination Protocol:*

In this paper, we present ACD, an Adaptive Code Dissemination Protocol for wireless sensor networks. Compared to prior works, ACD has four salient features. First, it supports dynamically configurable packet sizes. By increasing the packet size for high PHY rate radios, it significantly improves the transmission efficiency. Second, it employs an accurate sender selection algorithm to mitigate transmission collisions and transmission over poor links. Third, it employs a simple impact based back-off timer design to shorted the time spent in coordinating multiple eligible sender so that the largest impact sender is most likely to transmit. Fourth, it employs algorithms to handle undercover nodes to solve the hidden terminal problem. All this significantly increases the channel quality. We implement ACD and evaluate its performance extensively. Results show that ACD outperforms state-of –the-art protocols, deluge and MNP, in terms of completion time and data traffic.

ACD effectively improves the performance in terms of completion time and data transmission overhead. We enable dynamically configurable packet sizes to support large packets to improve the dissemination performance. We employ an accurate and fast sender selection algorithm to alleviate concurrent transmission and transmission over poor links. We employ a synchronization algorithm which will provide solution to hidden terminal problems occurs in undercover nodes.

B. *Sender Selection Algorithm:*

Sender Selection is a well studied technique to reduce contention and collision in broadcast protocols. It was also adopted in a previous code dissemination protocol MNP [5] for improved performance. The basic principle in these algorithms is to select the best sender for forwarding the data while avoiding simultaneous transmission from other neighboring nodes.

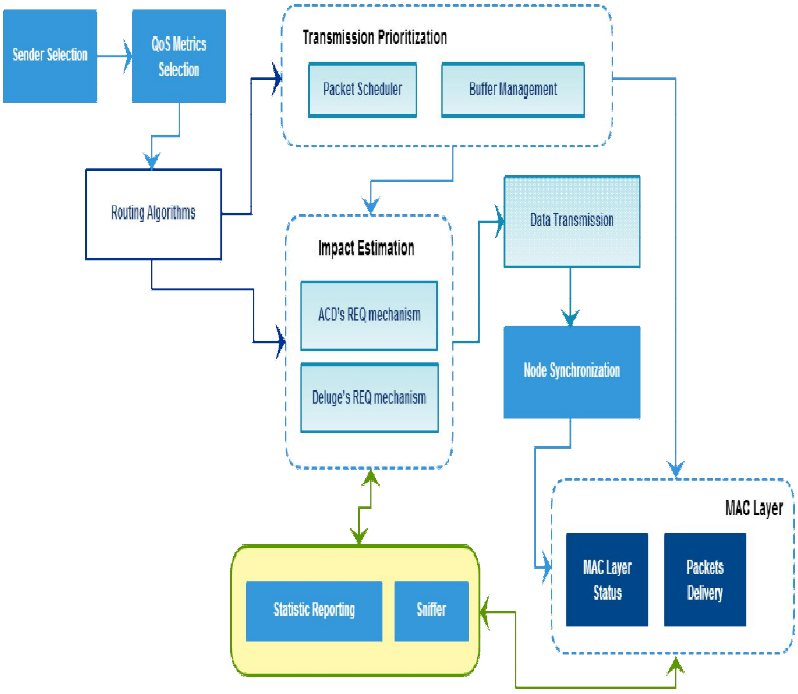
This involves two main aspects; first, an accurate metric should be devised to estimate senders' impacts. Second, efficient mechanisms should be designed to coordinate transmissions of eligible senders so that the largest impact sender is most likely to transmit. Sender selection in broadcast protocols is a special case of sender selection in code dissemination protocols (i.e., one page, one packet). Sender selection in code dissemination protocols is more complex.

In particular, code dissemination should be 100 percent reliable, so sender are enforced to receive ACK/NACK after transmission; neighboring nodes should determine whether a page transmission is in the vicinity because the packet transmission within a page are not continuous. MNP is a code dissemination protocol that incorporates sender selection. The metric it uses for sender selection is the received number of distinct requests. A sender with more request is more suitable for transmission because its transmission can cover a large number of receivers. The best sender is selected via multiple rounds. The existence of the best node makes other neighboring nodes inactive; then the receivers request the code from the active sender. The problem therein is that it can cause biased estimation of nodes' real impacts. Resulting in unnecessary packet transmissions.

C. *Architectural Diagram:*

Step 1: Sender Selection:

Sender Selection is a technique to reduce contention and collisions in broadcast protocols. It is used to select the best sender for forwarding the data avoiding simultaneous transmission from other neighbouring nodes. This involves two main aspects. First, an accurate metric should be devised to estimate senders' impacts. Second, efficient mechanisms should be designed to coordinate transmission of eligible sender so that the largest impact sender is most likely to transmit.



Step 2: Impact Estimation:

Impact Estimation is used to estimate the number of uncovered nodes and the outbound link qualities to them. To estimate the number of uncovered nodes, we use the REQ messages sent by uncovered nodes when missing packets are detected. Multiple eligible sender overhear the REQ messages and may be responsible for sending request packets in the REQ message that is not destined for them. This enlarges the set of eligible senders so that we select the best one.

Step 3: Transmission Prioritization:

All nodes that overhear the REQ messages and have the requested page are eligible sender. From these eligible senders, we want to select the best sender. ACD uses an implicit back-off mechanism which can significantly reduce the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit.

Step 4: Node Synchronization:

In our sender selection process, we implicitly synchronize (and align) the estimation period during which the impact is being estimated. Otherwise, there would be hidden terminal problems.

Step 5: Sniffer:

The sniffer component is installed to a separate node called the sniffer node. It is able to hear all transmission within the radio range. After the dissemination process, all other nodes report their statistics with the maximum transmission power so that the sniffer node can hear and transfer the results.

IV. Simulation Results

The simulation studies involve the deterministic small network topology with 5 nodes as shown in Fig.1. The proposed energy efficient algorithm is implemented with MATLAB. We transmitted same size of data packets through source node 1 to destination node 5. Proposed algorithm is compared between two metrics Total Transmission Energy and Maximum Number of Hops on the basis of total number of packets transmitted, network lifetime and energy

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

consumed by each node. We considered the simulation time as a network lifetime and network lifetime is a time when no route is available to transmit the packet. Simulation time is calculated through the CPUTIME function of MATLAB. Our results shows that the metric total transmission energy performs better than the maximum number of hops in terms of network lifetime, energy consumption and total number of packets transmitted through the network.

The network showed in Fig. 1 is able to transmit 22 packets if total transmission energy metric is used and 17 packets if used maximum number of hops metric. And the network lifetime is also more for total transmission energy. It clearly shows in Fig. 2 that the metric total transmission energy consumes less energy than maximum number of hops. As the network is MANET means nodes are mobile and they change their locations. After nodes have changed their location the new topology is shown in Fig .3 and energy consumption of each node is shown in Fig. 4. Our results shows that the metric total transmission energy performs better than the maximum number of hops in terms of network lifetime, energy consumption and total number of packets transmitted through the network.

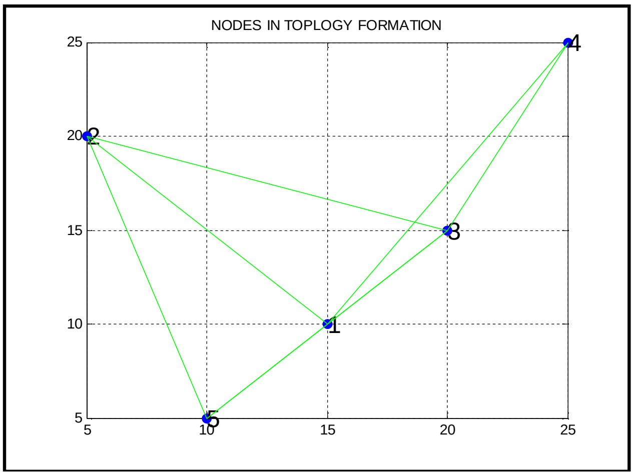


Fig.1. Ad Hoc Network of 5 Nodes

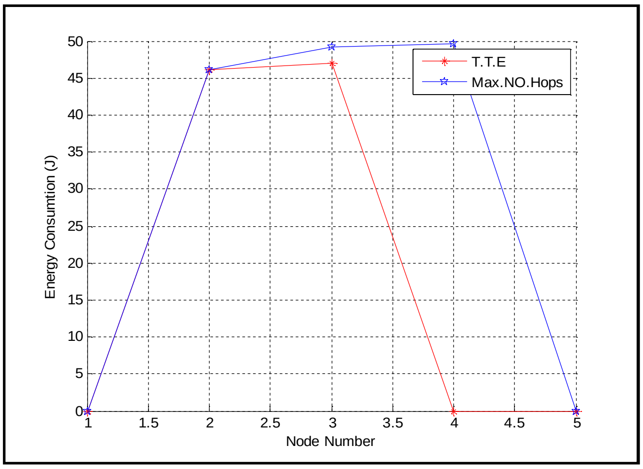
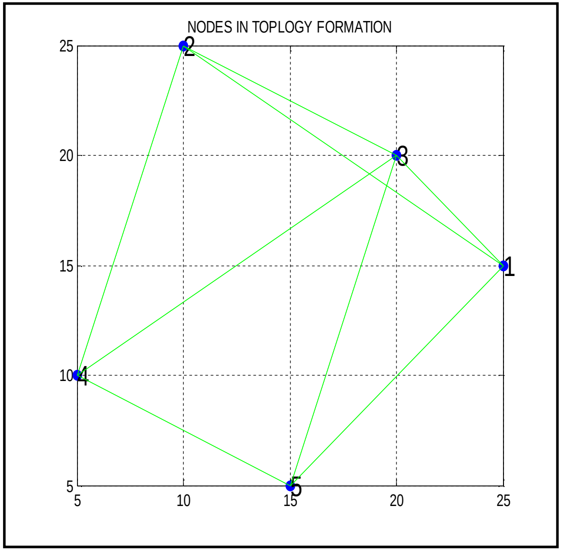
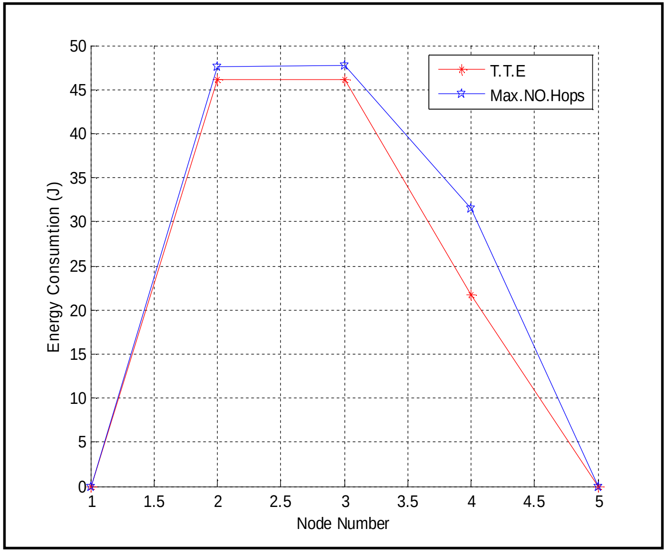


Fig. 2. Energy Consumption by Each Node





V. Conclusion and Future Work

In this paper, we present ACD, an Adaptive Code Dissemination protocol for wireless sensor networks. Compared to prior works, ACD has four salient features. First, it supports dynamically configurable packet sizes. By increasing the packet size for high PHY rate radios, it significantly improves the transmission efficiency. Second, it employs an accurate sender selection algorithm to mitigate transmission collisions and transmissions over poor links. Third, it employs a simple impact-based back off timer design to shorten the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit. Fourth, it employs algorithms to handle undercover nodes to solve the hidden terminal problem. All this significantly increases the Channel Quality. We implement ACD and evaluate its performance extensively. Results show that ACD outperforms state-of-the-art protocols; Deluge and MNP, in terms of completion time and data traffic and also solves the hidden terminal problems. In future we would like to examine effectiveness of our design in large scale Wireless Sensor Networks systems.

REFERENCES

1. Leijun Huang, Sanjeev Setia, ’CORD: Energy-Efficient Reliable Bulk Data Dissemination in Sensor Networks’, Proceedings of International Conference,1997
2. Priyadharsini R. Joseph K.S. ‘Handling hidden and exposed terminal problems using agent based access scheman’, Proceedings of International Conference on Intelligent Agent and Multi Agent System, page:1-6, July 2009.
3. W. Dong, X. Liu, C. Chen, Y. He, G. Chen, Y. Liu, and J. Bu, ‘DPLC: Dynamic Packet Length Control in Wireless SensorNetworks’, Proceedings of IEEE INFOCOM, pp. 1-9,2010.
4. J.W. Hui and D. Culler, ‘The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale’, in Procedings ACM Sensor System, pp. 81-94, 2004.
5. S. Kulkarni and L. Wang, ‘Energy-Efficient Multihop Reprogramming for Sensor Networks’, ACM Transmission Sensor Networks, vol. 5, no. 2, p. 16, March 2009.
6. T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, ‘Exploiting Link Correlation for Efficient Flooding inWireless Sensor Networks’, in Proceedings. USENIX Symp. NSDI, p. 4, 2010.
7. Tinos TEP 124: The Link Estimation Exchange Protocol (LEEP). [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep124.html

BIOGRAPHY

C. Bhuvaneswari is a Student in Department of Computer Science, Sir Issac Newton College of Engineering and Technology, Anna University. She worked as a Assistant Professor from 2010 - 2013 in R.V.S. College of Computer Applications, Karaikal, Puducherry. She received Master of Science in Information Technology (M.Sc.) degree in 2008 from Bharthidasan University, TamilNadu, India.