International Journal of Engineering Research and General Science Volume 3, Issue 4, Part-2, July-August, 2015
ISSN 2091-2730

Excellent Code Dissemination Protocol using Multi-objective Optimization in WSN

Saniya Elza Dominic, Gayathri.R.Krishna

MG university, Saniya.prakkadan@gmail.com, 9447124870

Abstract— Wireless reprogramming is a difficult technique for software deployment in wireless sensor networks (WSNs). Code dissemination is a basic building block to enable wireless reprogramming. Here present ECD, an Excellent Code Dissemination protocol leveraging 1-hop link quality information based on the TinyOS platform. Compared to previous works, ECD has three main features. First, it supports dynamically configurable packet sizes. By increasing the packet size for high PHY rate radios, it improves the transmission efficiency. Second, it has a precise sender selection algorithm to mitigate transmission collisions and transmissions over poor links. Third, it has a simple impact-based backoff timer design to shorten the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit. These features is incorporated into multi objective optimization (MOPs) approach using Genetic algorithm (GA) to further reduce delay due to best sender selection. Results show that ECD-MOP outperforms other protocols, in terms of completion time.

Keywords— Dissemination, Delay, Genetic algorithm, Link quality, Multi objective optimization, Reprogramming, Wireless sensor network.

I. INTRODUCTION

Recent development in microelectronic mechanical systems and wireless communication technologies have fostered the rapid development of networked embedded systems like wireless sensor networks (WSNs) [1], [23]. A wireless sensor network (WSN) consists of a number of nodes used for monitoring purposes which pass the information collected via the network to a main location.

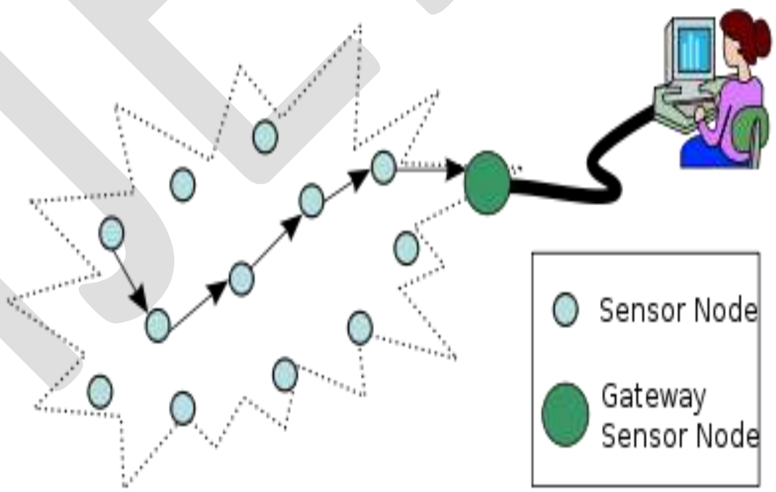

Fig.1. Example of a wireless sensor network

WSN applications often need to be changed after deployment for a variety of reason, reconfiguring a set of parameters, changing tasks of individual nodes, and patching security holes. Many large-scale WSNs [12], however, are deployed in environments where physically collecting previously deployed nodes is either very difficult or infeasible. Wireless reprogramming is a crucial technique to address such challenges.

Code dissemination is a basic building block for wireless reprogramming [4], [21]. Existing code disseminationprotocols (represented by Deluge [6] and MNP [7]) adopt several key techniques to ensure high reliability and performance. First, they exchange control-plane messages for high reliability [6], [7], and [21]. Second, they segment a large code object into fixed-sized pages for pipelining [6], [7]. 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 inefficiency in two main areas. 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 [7] (for addressing the broadcast storm problem [14]) does not consider link quality information and needs multiple rounds of message exchanges, resulting in transmission redundancy and long completion time. To address the first issue, here increase the packet size to improve the transmission efficiency for high PHY rate radios. However, it would be inappropriate to fix the packet size to its maximum allowable size as a fixed packet size may not be appropriate for all platforms under every conditions [3]. Therefore, we support dynamically configurable packet sizes in design.

To overcome the second issue, leverges 1-hop neighbors' link quality information learned over the air to improve the sender selection accuracy. Here dynamically estimate the impacts of senders by considering both uncovered neighbours (i.e., neighbors that do not receive an entire page) and the link qualities to those neighbors. A node's transmission is considered more effective if the node has more uncovered neighbours with good link qualities. Taking link qualities help to put minimum weight on senders with poor link qualities to their neighbors. This is especially important when large packets are transmitted over the air. The basic idea is to prioritize sender transmissions so that the best sender with the largest impact is most likely to transmit.

Here an impact-based backoff timer design is used to shorten the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit. To further reduce delay due to accurate sender selection, Here incorporate Multi objective based optimization using genetic algorithm. Genetic algorithm is known as a global heuristic algorithm, a genetic algorithm generates an optimal solution through generating different individuals. Focused fitness function is one of procedures of the algorithm.

II. RELATED WORKS

Deluge [6] is perhaps the most popular code dissemination protocol used for reliable code updates. Hui et al. gave Deluge which is a reliable data dissemination protocol for propagating large data objects (by dividing those to fixed sized pages) from one source node to other nodes over a multi-hop, wireless sensor network [6]. Dissemination of large data objects i.e. program images poses many issues like large size of programs, toleration of varying node densities and ensuring complete reliability in transfer etc.

Deluge achieves reliability in unpredictable wireless environments and robustness when node densities can vary by factors of a thousand or more. This protocol is based on Trickle algorithm. Here each and every node follows a set of strictly local rules to achieve data dissemination in the network. A node at regular intervals advertises the most recent version of the data item it has to whichever nodes that can hear its broadcast. Consider B receives an advertisement from an older node A, and then B will respond with the information that it has. From the information received, A determines which portion of the data items need updating and requests them from any neighbour that advertises the availability of the needed data, including B. Nodes receiving these requests then broadcast any requested data. Thus nodes advertise newly received data in order to propagate it further to other nodes. A problem in Deluge is that when a sender receives requests from receivers, it will start transmitting data packets after a specified timeout. Multiple senders in a neighborhood start transmitting concurrently, causing collisions.

Sandeep et al. proposed a multihop network reprogramming protocol (MNP) [7]. It provides a reliable service to propagate new program code to all sensor nodes in the network. The main aim of this dissemination protocol is to ensure reliable, low memory usage and fast data dissemination. It is based on a sender selection protocol in which source nodes compete with each other based on the number of distinct requests they have received. In each neighbourhood, a source node sends out program codes to multiple receivers. When the receivers get the whole program image at their side, they become source nodes, and send the code into their neighbourhood. But here there can be issues of collisions. This is solved by selecting a suitable sensor node based on some parameters maintained by the nodes and some advertisement and download messages exchanged by the nodes. It is like a greedy algorithm. Pipelining can be used in this protocol to enable faster data propagation in the case of larger networks. To do pipelining, programs are divided into segments, each of which contains a fixed number of packets. Once a sensor node receives all the segments of a program, it can reboot with the new program. This continues till all the nodes are hence updated.

Compared to the above two works, our system has two main differences. First, a dynamically configurable packet sizes to support large packets to improve the dissemination performance. Second, employs an accurate and fast sender.Sprinkler [13] uses the localization service at each node to construct a connected dominating set (CDS). It uses TDMA to schedule packet transmissions among the CDS nodes to reduce energy consumption by minimizing packet transmissions. ECD mitigate the problem by limiting the number of concurrent transmitters in a neighborhood via dynamic sender selection.. Sprinkler, employs a TDMA-based approach. Sprinkler assumes a unit disk radio model. Nodes can be assigned the same time slot (and can concurrently transmit) only when the distance is sufficiently large However, whether there are collisions in practice depends on whether the unit disk radio model is accurate enough as well as whether the model keeps unchanged over time.

CF [22] is a recent work that exploits spatial link correlation to mitigate ACK overhead. ECD vary with CF in three aspects. First, CF is used for flooding a single packet while ECD is used for disseminating large code objects consisting of multiple pages and packets. CF does not assure a100 percent reliability while ECD employs handshake and negotiation to achieve 100 percent reliability. Second, the basic rule for sender selection are the same for both protocols, i.e., choosing the sender which can cover the most number of uncovered (i.e., not yet received) receivers in a neighborhood. The techniques used in both protocols, however, are different. CF relies on ACKs and exploits link correlation to estimate the expected number of uncovered nodes while ECD relies on NAKs (i.e., REQ messages) and the link qualities to estimate the expected number of uncovered receivers. Third, both protocols rely on backoff timers to prioritize the transmissions. In CF the backoff period is simply calculated as the reciprocal of the impact. In ECD, the backoff period is more carefully optimized by minimizing the probabilities of ‘‘priority inversion’’ and transmission collisions.

III. MOTIVATION

This section identifies the need to incorporate a more accurate sender selection algorithm.

Sender Selection

Sender selection is a well studied technique to reduce contentions and collisions in broadcast protocols. The basic principle in these algorithms is to select the best sender for forwarding the data while avoiding simultaneous transmissions from other neighboring nodes. This includes two main aspects. First, an precise metric should be used 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.

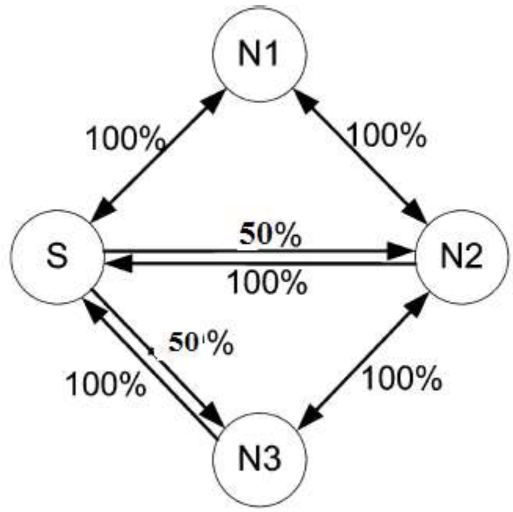

Fig. 2 Example of sender selection.

Fig. 2 shows the benefits of our accurate sender selection algorithm. Assume S is the source node and there are 20 packets per page. In Fig. 2, indicate the link qualities by the figures above the edges. For example, the link quality from S to N2 is 50 percent while the reverse link quality is 100 percent. Therefore, S can directly communicate with N1, N2, N3. However, the link qualities are different. Since the link quality from S to N1 is 100 percent, after S's transmission of a page, N1 can receive all the packets. On the other hand, since the link qualities from S to N2 and N3 are 50 percent, N2 and N3 can thus receive 50% packets. With MNP's approach, S will still be the next sender because it will receive the most number of requests. Hence, in MNP, 80 transmissions are needed for S to cover all N1, N2, N3. ECD considers link qualities. By one round of page transmission, S cannot cover one receiver whereas N1 can cover N2. Therefore, N1 will be the sender (covering N2 by one page transmission), followed by N2 being the next sender (covering N3 by one page transmission).These cause delay in ECD.Because higher link quality data reaches slower than lower link quality data. Through higher link quality link more data will transmit than lower. This cause extra delay.

Impact Estimation

MNP's sender selection algorithm may spend too many transmissions on poor links. It can be avoided in our protocol design. Here ECD estimate the number of uncovered nodes and the outbound link qualities to them. To calculate the number of uncovered nodes, use the REQ messages sent by uncovered nodes when missing packets are detected. There are two differences between ECD's REQ mechanism and Deluge's REQ mechanism. First, one or more eligible senders overhear the REQ message and may be responsible for sending requested packets in the REQ message that is not destined for them. This makes the set of eligible senders so that ECD select the best one. Second, it is to be note that in Deluge [6], REQ messages may be suppressed if another REQ message for the same page is overheard. This mechanism, however, will lead to biased estimation in protocol design. Because of this reason, in ECD, the uncovered nodes send REQ messages unless there is an ongoing page transmission.

The ability to accurately estimate wireless link quality is critical to the performance of routing protocols. Link quality estimation has been an important research topic in the wireless sensor networking community and researchers have developed a large number of different methods to estimate link quality. Link quality prediction is an important approach to solve this problem. Here ECD estimate the link quality based on the past knowledge and information, link quality prediction is essential for routing decisions for future data transmission

IV. MULTI OBJECTIVE OPTIMIZATION FORMULATION

Although substantial amount of search in optimization is conducted with regards to single objective problems, optimization problems with multi conflicting objectives are inevitable in many topics specially engineering applications. Two main methods have been proposed by scientist for solving multi-objective optimization problems: 1) Classical method, 2) Evolutionary algorithms. Classical methods are able to reach one optimal solution at each run, while evolutionary algorithms are based on a population of solutions which will hopefully lead to a number of optimal solutions at every generation. The evolutionary algorithm method which had shown benefits over the classical approach can be categorize in to varies categories. Genetic Algorithm is one of the methods that imitate the evolution of genes and chromosomes.

The aim is to develop a fast and efficient multi-objective optimization technique by using GA (Genetic Algorithm) method, in order to solve multi-objective optimization problems with constraints. When only one objective function involves in the problem, it is called single objective optimization, however in most real world problems more than one objective function is required to be optimized, and therefore these problems are named multi-objective optimization. This paper tries to reduce delay to maximize the performance and reduce cost of routing using optimization technique.

The characteristic of evolutionary methods which use a population of solutions that evolve in each generation is well suited for multi-objective optimization problems. Since one of the main goals of MOOP solvers is to find a set of non-dominated solutions with the minimum distance to pareto-front, evolutionary algorithms can generate a set of non-dominated solutions in each generation. The first goal in multi-objective optimization is achieved by a proper fitness assignment strategy and a careful reproduction operator. Diversity in the pareto-set can be obtained by designing a suitable selection operator.

Consider a decision-maker who wishes to optimize K objectives such that the objectives are non-commensurable and the decision-maker has no clear preference of the objectives relative to each other. Without loss of generality, all objectives are of the minimization type a minimization type objective can be converted to a maximization type by multiplying negative one. A minimization multi-objective decision problem with K objectives is defined as follows: Given an n-dimensional decision variable vector $\mathbf{x}=\{x_1,\ldots,x_n\}$ in the solution space $\mathbf{X}$, find a vector $\mathbf{x}^*$ that minimizes a given set of K objective $z(x^*)=\{z_1(\mathbf{x}^*),\ldots,z_K(\mathbf{x}^*)\}$[23]. The solution space $\mathbf{X}$ is generally restricted by a series of constraints, such as $g_j(\mathbf{x}^*)=b_j$ for $j=1, \ldots, m$, and bounds on the decision variables[23].

In many real-life situations, objectives under consideration conflict with each other. Hence, optimizing value $\mathbf{x}$ with respect to a single objective often results in unexpectable results with respect to the other objectives. Therefore, a best multi-objective solution that simultaneously optimizes each objective function is almost impossible. A best solution to a multi-objective problem is to find a set of solutions, each of which satisfies the objectives at an normal level without dominating one over other solution.

If all objective functions are for minimization, a good solution $\mathbf{x}$ is said to dominate another good solution $\mathbf{y}$ ($\mathbf{x} \succ \mathbf{y}$), if and only if, $z_i(\mathbf{x})\leqslant z_i(\mathbf{y})$ for $i=1, \ldots, K$ and $z_j(\mathbf{x})<z_j(\mathbf{y})$ for least a objective function j. A resultant solution is said to be *Pareto optimal* if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be further changed with any aspect to any other objective without worsening at least one other objective. The set of all feasible non-dominated solutions in $\mathbf{X}$ is referred to as the *Pareto optimal set* [21].

The final goal of a multi-objective optimization algorithm is to find an solutions in the Pareto optimal set. However, finding such an Pareto optimal set, for many multi-objective problems, is practically not possible due to its bigger size. In addition, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Therefore, a way to multi-objective optimization is to investigate a set of solutions that represent the Pareto optimal set as well as possible.

Genetic Algorithm

The concept of GA was developed by Holland and his colleagues in the 1960s and 1970s. GA was inspired by the evolutionist theory explaining the origin of species. In nature, weak and unfit species within environment are faced with extinction by natural selection [22]. The fit ones have greater opportunity to pass their genes to future generations via reproduction. Species carrying the right combination in their genes become dominant in their population. In long process, of evolution, random changes may occur in genes. If these changes provide additional benefit in the challenge for survival, new species develop from the old ones.

In GA terminology, a solution vector $\mathbf{x}\in\mathbf{X}$ is called an individual or a *chromosome.* Chromosomes are made of number of units called *genes [22]*. Each gene controls one or more features of the chromosome. In the implementation of GA by Holland, genes are assumed to be binary digits. In later implementations, more varied gene types have been introduced. A mapping mechanism between the solution space and the chromosomes is required. This mapping is called an encoding. GA operates with a collection of chromosomes, called a *population*. The population is randomly initialized. As the search evolves, the population contains fitter and fitter solutions, and eventually it converges, result is a single dominated solution. Holland also presented a proof of convergence (the schema theorem) to the global optimum where chromosomes are binary vectors.

GA uses two operators to generate new solutions from existing ones: *crossover* and *mutation*. The crossover operator is the mostly used operator of GA. In crossover, two chromosomes, called *parents*, are joined together to form new chromosomes, called *offspring*. The parents are selected among already existing chromosomes in the population pool with preference towards fitness so that offspring is expected to get a good genes which make the parents fitter. By repeatedly applying the crossover operator, genes of best chromosomes are expected to appear more frequently in the population, which leads to convergence to an overall good solution.

The mutation operator introduces sudden and random changes into characteristics of chromosomes. Mutation is generally applied at the gene. In GA implementations, the mutation rate (probability of changing the properties of a gene) is very low and depends on the length of the chromosome. Therefore, the new chromosome produced by mutation will not have a greater difference from the original one. Mutation plays a crucial role in GA.

Reproduction involves selection of chromosomes for the next generation. In the most general case, the fitness of an individual determines the probability of its survival for the next generation. There are varies selection procedures in GA depending on how the fitness values are used. Proportional selection, ranking, and tournament selection are the most popular selection procedures. The procedure of a generic GA is given as follows:

Step 1: Set $t=1$. Randomly generate N solutions to form the first population, P_1. Evaluate the fitness of solutions in P_1.

Step 2: *Crossover*: Develop an offspring population N_t as follows:

2.1. Choose two solutions **x** and **y** from P_t based on the fitness values.

2.2. Using a crossover operator, generate offspring and add them to N_t.

Step 3: *Mutation:* Mutate each solution $\mathbf{x} \in N_t$ with a predefined mutation rate.

Step 4: *Fitness assignment:* Evaluate and assign a fitness value to each solution $\mathbf{x} \in N_t$ based on its objective function value and infeasibility.

Step 5: *Selection:* Select N solutions from N_t based on their fitness and copy them to P_{t+1}.

Step 6: If the stopping criterion is satisfied, stop the search and return to the current population, else, set t=t+1 go to Step 2.

Multi-objective GA

Being a population-based approach, Genetic algorithm is mostly opted to solve multi-objective optimization problems. A single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems. In addition, most multi-objective GA do not require the user to prioritize, scale, or weigh objectives. Therefore, GA has been the most popular heuristic approach to multi-objective design and optimization problems. Jones et al. reported that 90% of the approaches to multi-objective optimization aimed to approximate the true Pareto front for the underlying problem.

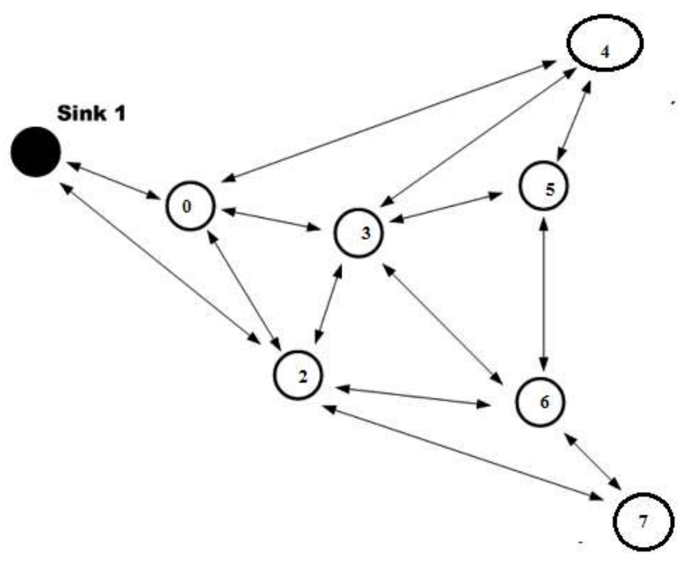

Fig. 3 Code dissemination nodes

Here in a network, select 1 as sink node to propagate code dissemination. Based on ECD-MOP each node under goes initialization, selection, mutation and crossover. This process goes for 48 rounds. Until best solution is obtained. Solution so obtained in ECD-MOP has least delay and Fast completion time.

0	1	2	3	4	5	6	Rand1	Rand2	Rank
4	6	10	6	10	1,2	0	0	0	4.330275...
4	2,0,6	10	4	10	1,2,3	4	0	0	2.632520...
4	6,0,3,2	4	6	10	3	0,4	0	0	4.888888...
4	2,6,0,3	10,4	4,6	10	2,1,3	4	0	0	2.885527...
4	6,3	10	4	10	2,3	4	0	0	2.708333...
4	0,2	4,10	6	10	3	0,4	0	0	4.888888...
4	3	10	4	10	2,3,1	0	0	0	3.266666...
4	0,3	10,4	4,6	10	1	4,0	0	0	4.745896...
4	6,3,2,0	10	4,6	10	2,3	4	0	0	2.618861...
4	3,6,2	10	6	10	2,3,1	0	0	0	4.662015...

Fig.4 Initialization phase on above code dissemination node

The above table shows how initialisation is done on each node. After initialisation, selection is next phase. For each node select combination. Then crossover and mutation is done. This is continued in several rounds. Finally obtain best solution.

Performance Evaluation

In this section, we evaluate the performance of ECD with multi objective optimization using GA through one's simulation. Here first describe our performance metrics and simulation scenarios. Then evaluate the system performance with given scenarios and parameters. Finally, done comparisons between our ECD and ECD-MOP. The results confirm that ECD-MOP (multi objective optimization) have fast completion time, less delay.

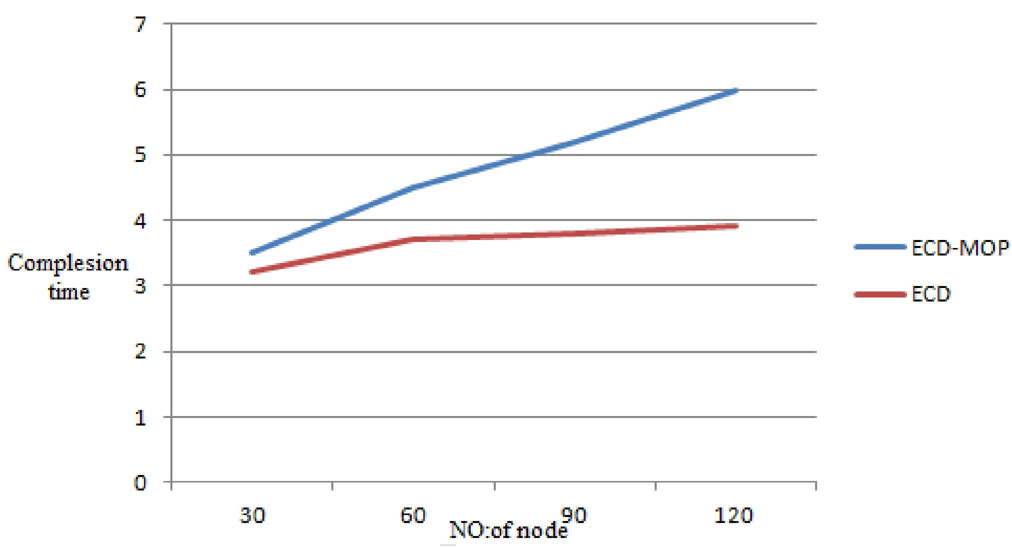

Fig .5 Performance evaluations on completion time

Acknowledgment

The first author would like to thanks all those people, who guided and supported. Without their valuable guidance and support, this task was not possible and also likes to thank colleagues for their discussions and suggestions.

CONCLUSION

Here present ECD, an Excellent Code Dissemination protocol for wireless sensor networks. Compared to prior works, ECD has three main 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 has an accurate and best sender selection algorithm to mitigate transmission collisions and transmissions over poor links. Third, ECD has a simple impact-based backoff timer design to shorten the time spent in coordinating multiple eligible senders so that the largest impact sender is most likely to transmit. In order to further reduce delay and overhead due to best sender selection, here incorporate multi objective optimization technique using Genetic Algorithm (ECD-MOP). This will provide a solution which has less delay and Fast completion time.

REFERENCES:

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ‘‘Wireless Sensor Networks: A Survey,’’ Comput. Netw., vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] W. Dong,C.Chen, X. Liu, J. Bu, and Y. Liu, ‘‘Performance of Bulk Data Dissemination in Wireless Sensor Networks,’’ in Proc. IEEE/ ACM Int’l. Conf. DCOSS, 2009, pp. 356-369.

[3] W. Dong, X. Liu, C. Chen, Y. He, G. Chen, Y. Liu, and J. Bu, ‘‘DPLC: Dynamic Packet Length Control in Wireless Sensor Networks,’’ in Proc. IEEE INFOCOM, 2010, pp. 1-9.

[4] W. Dong, Y. Liu, X. Wu, L. Gu, and C. Chen, ‘‘Elon: Enabling Efficient and Long-Term Reprogramming for Wireless Sensor Networks,’’ in Proc. ACM SIGMETRICS, 2010, pp. 49-60.

[5] M. Heusse, F. Rousseau, R. Cuillier, and A. Duda, ‘‘Idle Sense: An Optimal AccessMethod for High Throughput and Fairness in Rate Diverse Wireless LANs,’’ in Proc. ACM SIGCOMM, 2005, pp. 121-132.

[6] J.W. Hui and D. Culler, ‘‘The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale,’’ in Proc. ACM SenSys, 2004, pp. 81-94.

[7] S. Kulkarni and L. Wang, ‘‘Energy-Efficient Multihop Reprogramming for Sensor Networks,’’ ACM Trans. Sens. Netw., vol. 5, no. 2, p. 16, Mar. 2009.

[8] P. Levis, N. Lee, M. Welsh, and D. Culler, ‘‘TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications,’’ in Proc. ACM SenSys, 2003, pp. 126-137.

[9] P. Levis, N. Patel, D. Culler, and S. Shenker, ‘‘Trickle: A Self- Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks,’’ in Proc. USENIX Symp. NSDI, 2004, p. 2.

[10] Y. Liu, Y. He, M. Li, J.Wang, K. Liu, L. Mo, W. Dong, Z. Yang, M. Xi, and J. Zhao, ‘‘Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs,’’ in Proc. IEEE INFOCOM, 2011, pp. 873-881.

[11] W. Lou and J. Wu, ‘‘Towards Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage,’’ IEEE Trans. Mobile Comput., vol. 6, no. 2, pp. 148-163, Feb. 2007.

[12] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X.-Y. Li, and G. Dai, ‘‘Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest,’’ in Proc. ACM SenSys, 2009, pp. 99-112.

[13] V. Naik, A. Arora, P. Sinha, and H. Zhang, ‘‘Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Wireless Embedded Devices,’’ in Proc. IEEE RTSS, 2005, p. 286.

[14] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, ‘‘The Broadcast Storm Problem in a Mobile Ad Hoc Networks,’’ in Proc. ACM MobiCom, 1999, pp. 151-162.

[15] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. Mclaughlin, ‘‘Energy Efficiency Based Packet Size Optimization in Wireless Sensor Networks,’’ in Proc. IEEE Int’l. Workshop Sens. Netw. Protocols Appl., 2003, pp. 1-8.

[16] S. Sen, N. Santhapuri, R.R. Choudhury, and S. Nelakuditi, ‘‘AccuRate: Constellation Based Rate Estimation in Wireless Networks,’’ in Proc. USENIX Symp. NSDI, 2010, p. 12.

[17] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, ‘‘Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks,’’ Stanford Univ., Stanford, CA, USA, 2006, Tech. Rep. SING-06-00.

[18] F. Stann, J. Heidemann, R. Shroff, and M.Z. Murtaza, ‘‘RBP: Robust Broadcast Propagation in Wireless Networks,’’ in Proc. ACM SenSys, 2006, pp. 85-98.

[19] TinyOS TEP 124: The Link Estimation Exchange Protocol (LEEP). [Online].Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep124.html

[20] M. Vutukuru, H. Balakrishnan, and K. Jamieson, ‘‘Cross-Layer Wireless Bit Rate Adaptation,’’ in Proc. ACM SIGCOMM, 2009, pp. 3-14.

[21] http://www.sciencedirect.com/science/article/pii/S0951832005002012

[22] http://www.ijsce.org/attachments/File/v3i2/B1423053213.pdf

[23] **http://www.researchgate.net/publication/269330296**