

MuMHR: *Multi-path, Multi-hop Hierarchical Routing*

Mohammad Hammoudeh*, Alexander Kurz†, and Elena Gaura*

*Cogent Computing Applied Research Centre,
Department of Creative Computing
Coventry University
Coventry, CV1 5FB, UK. Email: aa2792@coventry.ac.uk
†Department of Computer Science
University of Leicester
Leicester, LE1 7RH, UK. Email: kurz@mcs.le.ac.uk

Abstract— This paper proposes a self-organizing, cluster based protocol- *Multi-path, Multi-hop Hierarchical Routing (MuMHR)* - for use in large scale, distributed Wireless Sensor Networks (WSN). With MuMHR, robustness is achieved by each node learning multiple paths and election of cluster-head backup node(s). Energy expenditure is reduced by shortening the distance between the node and its cluster-head and by reducing the setup communication overhead. This is done through incorporating the number-of-hops metric in addition to the back-off waiting time. Simulation results show that MuMHR performs better than LEACH, which is the most promising hierarchical routing algorithm to date; MuMHR reduces the total number of set-up messages by up to 65% and enhances the data delivery ratio by up to 0.83.

I. INTRODUCTION

Hierarchical routing is one of the most popular routing schemes in sensor networks [4], [8], [10], [11], [14], [15], [17]. It is a two or more tier routing scheme known for its scalability and communication efficiency. Nodes in the upper tier are called cluster-heads and act as a routing backbone, while nodes in the lower tier perform the sensing tasks. Kulkarni et al. [6] argue that multi-tier networks are scalable and offer a number of advantages over single-tier networks: lower cost, better coverage, higher functionality, and better reliability. A sink-based single tier network can lead to congestion at the gateway especially in dense sensor networks. This can cause communication delays and inadequate tracking of the sensed events. Moreover, some of the routing algorithms for such architecture are commonly not scalable. To overcome these problems, network clustering has been proposed as a possible solution.

Low Energy Adaptive Clustering Hierarchy (LEACH) [5] is one of the most promising routing algorithms for sensor networks [4], [8], [10], [11], [14], [15], [17]. However, LEACH has been based on a number of assumptions which in the authors' opinion limit its effectiveness in a number of applications. This paper proposes modifications to LEACH which will improve the robustness of the algorithm and reduce the energy consumption in the network. The dynamic clustering brings extra overhead, such as head changes, which may diminish savings in energy consumption. A possible solution

which is examined in this paper in order to reduce set-up communication overhead is the use of the back-off time for advertisement messages. Also the assumption that every node can transmit to reach the sink is relaxed by enabling multi-hop transmissions. Finally, to add reliability to the protocol, nodes transmit over multiple paths transmissions and backup cluster-heads are elected.

The rest of the paper is organized as follows. Section II briefly describes the network model and assumptions. Section III reviews the related work. Section IV describes the simulation tool used for the work in this paper. Section V exhibits the details of MuMHR. Section VI presents simulation results. Section VII concludes the paper.

II. LEACH

The operation of LEACH is split into two phases, the set-up phase and the steady-state phase. To minimize overhead, the duration of the steady-state phase is longer than the set-up phase [5]. During the set-up phase, the clusters are created and cluster-heads are elected. This election is made by every node choosing a random number between 0 and 1. The sensor node becomes cluster-head if this random number is less than the threshold $T(n)$ set as:

$$T(n) = \begin{cases} P / (1 - P \times (r \bmod \frac{1}{P})) & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

where P is the desired percentage of cluster-heads, r is the current round and G is the set of nodes that have not been selected as a cluster-head in the last $1/P$ rounds. The desired percentage of cluster-heads was found to be 5% of the total number of nodes in the network [5]. Note that 0 cluster-heads and 100% cluster-heads is equivalent to direct communication.

After cluster-heads are chosen, they broadcast an advertisement message to the entire network declaring that they are the new cluster-heads. Every node receiving the advertisement decides to which cluster they wish to belong based on the signal strength of the received message. The sensor node sends a message to register with the cluster-head of their choice. Based on a Time Division Multiple Access (TDMA) approach,

the cluster-head assigns each node registered in its cluster a time slot to send its data. This requires that every node must support TDMA. The cluster-head keeps a list of all nodes in the cluster to inform them of the TDMA schedule.

During the steady-state phase, sensor nodes can start transmitting data to their respective cluster-head. The cluster-head applies aggregation functions to compress the data before transmission to the sink. After a predetermined period of time spent on the steady-state phase, the network enters the set-up phase again and starts a new round of creating clusters.

LEACH is well-suited for applications where constant monitoring is needed and data collection occurs periodically to a centralized location [7]. It increases network lifetime in two ways. First, the load is distributed to all nodes but not all at the same time. Second, there is lossless aggregation of data by the cluster-heads. The protocol is powerful and simple since nodes do not require global knowledge or location information to create clusters. LEACH is able to increase the network lifetime and it achieves a more than 7-fold reduction in energy dissipation compared to direct communication [5].

Despite the significant overall energy savings, however, the assumptions made by the protocol raise a number of issues:

- 1) LEACH assumes that all nodes begin with the same amount of energy and that the amount of energy a cluster-head consumes is more than that of a non-cluster node. It also assumes that the amount of energy consumed by cluster-heads in every cluster round is constant. This assumption is however not realistic. Furthermore, making adjustments for differences in energy consumption causes LEACH to be less efficient.
- 2) LEACH assumes that all nodes can communicate with each other and are able to reach the sink. Therefore, it is only suitable for small size networks.
- 3) LEACH requires that all nodes are continuously listening. This is not realistic in a random distribution of the sensor nodes, for example, where cluster-heads would be located at the edge of the network.
- 4) LEACH assumes that all nodes have data to send and so assign a time slot for a node even though some nodes might not have data to transmit.
- 5) LEACH assumes that all nearby nodes have correlated data which is not always true.
- 6) Finally, there is no mechanism to ensure that the elected cluster-heads (P) will be uniformly distributed over the network. Hence, there is the possibility that all cluster-heads will be concentrated in one part of the network.

III. RELATED WORK

A number of enhancements over LEACH have been proposed previously [4], [8], [10], [11], [14], [15], [17]. Lindsey and Raghavendra devised a protocol called Power Efficient GATHERing in Sensor Information Systems (PEGASIS) [14] that is an improvement over LEACH. As opposed to LEACH, PEGASIS has no clusters, instead it creates chains from sensor nodes so that each node communicates only with their closest neighbours and only one node is selected from the chain

to communicate with the sink. PEGASIS has a number of drawbacks [3]:

- It requires dynamic adjustment of network topology to route data, which introduces significant overhead.
- It requires location information.
- Similar to LEACH, PEGASIS assumes that all nodes can communicate with the sink directly.
- The head of the chain can become a bottleneck and cause excessive transmission delays.
- It assumes that all nodes start with the same level of energy and consumption rates are equal.

Hierarchical-PEGASIS [8] extends PEGASIS by introducing a hierarchy in the network topology. It aims to reduce transmission delays to the sink and proposes a data gathering scheme that balances the energy and delay cost. Although Hierarchical-PEGASIS avoids the clustering overhead, it still requires dynamic topological adjustments.

Manjeshwar and Agrawal implemented the Threshold-sensitive Energy Efficient Protocol (TEEN), that utilizes a hierarchical approach along with a data centric mechanism [10]. The same authors also developed the Adaptive Threshold-sensitive Energy Efficient sensor Network protocol (APTEEN) [11], which enhances TEEN by capturing periodic data collection and reacting to time critical events. They demonstrated that APTEEN performance is between LEACH and TEEN in terms of energy dissipation and network lifetime. TEEN gives the best performance since it decreases the number of transmissions. Both TEEN and APTEEN protocols require additional traffic control to continually update the threshold values and complexity of forming clusters in multiple levels, implementing threshold-based functions and dealing with attribute-based naming of queries.

Smaragdakis et al. [15] address the issue of heterogeneity (in terms of energy) of nodes. Their protocol, called SEP (Stable Election Protocol), is based on random selection of cluster-heads weighted according to the remaining node energy. This approach addresses the problem of varying energy levels and consumption rates but still assumes that the sink can be reached directly by all nodes.

LEACH-C (LEACH Centralised) [4] uses a central algorithm to form clusters. This algorithm is not robust since it requires location information for all sensors in the network.

LEACH and its derivatives have been successful in reducing the energy per bit required by each node and the network as a whole to communicate from the nodes to the sink. Nonetheless, most are built upon the inflexible assumptions that: every node is able to communicate directly with the sink; every communication path is equally likely to succeed; and, every node has the same starting energy level and uses energy at the same rate. This paper provides a protocol with the same underlying benefit as LEACH and derivatives but provides for multi-hop communication, and increases robustness by using multiple communication paths. Also, in comparison to LEACH and most derivatives, this protocol reduces the number of set-up messages required, and thus should extend network life.

IV. SENSORPLUS

In order to implement the routing algorithm proposed in this paper and study its properties, a new version of an in-house sensor network simulator called SenSorPlus was used. SenSor [12] is a realistic and scalable Python based simulator that provides a workbench for prototyping algorithms for WSN. It consists of a fixed API, with customisable internals. Each simulated sensor node runs in its own thread and communicates using the same protocols as its physical counterpart would be. This enables experimentation with different algorithms for managing the network topology, simulating fault management strategies and so on, within the same simulation. SenSorPlus is an extension of SenSor with an added interface between the simulation environment and different hardware platforms, for example the *Gumstix* [2] platform. SenSorPlus bridges between SenSor and the Gumstix to allow applications implemented within the simulator to be ported directly on to the hardware. Sensors are modelled using a pool of concurrent, communicating threads. Individual sensors are able to:

- 1) Gather and process data from a model environment
- 2) Locate and communicate with their nearest neighbours
- 3) Determine whether they are operating correctly and act accordingly to alter the network topology in case of faulty nodes being detected.

Separate interfaces gather information from the network and display it on the graph pane or the chart pane, where individual data can be plotted during the simulation. This partitioning allows us to experiment with different ways of processing individual node data into information.

V. MUMHR: *Mu*LTi-PATH, *Mu*LTi-HOP HIERARCHICAL ROUTING

In this section, the properties of the proposed routing protocol, MuMHR are described. The main objective of this protocol is to provide substantially energy-efficient and robust communication. The energy efficiency is achieved by load balancing at two levels: (1) Network level, which involves traffic multiplexing over multiple paths; (2) Cluster level, introducing rotation of the cluster-heads every given interval of time. This prevents energy depletion resulting from constantly using the same path for transmission or particular nodes being cluster-heads for a long duration. The multi-paths aspect is not only used for load balancing but also when path failures occur. When a path fails, an alternative path can be immediately used which allows the protocol to dynamically adapt to failures without delays or degradation in the quality of service. At the cluster set-up time, one or more nodes are chosen as cluster-head backup node(s). Backup cluster-head node substitute for the cluster-head in some failure cases or when the current cluster-head decides to reduce its participation in the protocol if its energy level approaches a certain threshold value. For instance, if the current cluster-head decides to hand its role to the backup node, it notifies the respective node and forwards to it necessary information, such as the backup nodes list, to avoid a complete cluster set-up phase.

In this protocol, the “number-of-hops” metric was introduced. It indicates how far the cluster-head is from the sensing node. This allows nodes to: (1) Select the nearest cluster-head node, which saves energy and reduces messaging needed to bridge the distance between the cluster-head and the sensor node; (2) Allows a node to learn the shortest path to the selected cluster-head. We have also used a back-off waiting time similar to one proposed in [16] to decrease the number of set-up messages and aid the formation of more geographically uniform clusters. During the back-off waiting time, sensor nodes receive advertisement messages and only consider the message with the smallest number-of-hops received during that time. This allows blocking of the advertisement flooding at the edges of neighbouring clusters. Both the number-of-hops metric and the back-off waiting time allow energy efficient cluster formation.

The operation of the proposed routing protocol can be split into two phases: the setup phase and the data transfer phase.

A. *Set-up Phase*

During the set-up phase cluster-heads are selected and clusters are created. We assume that a simple addressing scheme is in place and the sink knows the range of addresses of nodes in the network. The sink randomly selects 5% of the nodes as cluster-heads and broadcasts this information. Every node that receives the discovery message changes its state from “waiting” to “discovered” and examines the message to check whether it has been selected as cluster-head or not. If yes, it starts a new cluster by broadcasting an advertisement message. Otherwise, it forwards the message to its neighbours. Every node will remember the node from which it has received the discovery message as the immediate neighbour nearest to the sink. This path will be used only in cluster failure situations. Every cluster-head will create an advertisement message that has the number-of-hops parameter set to zero and broadcasts it to its neighbours. Upon receiving an advertisement message, a sensor node will do the following: (a) If the node already belongs to a cluster, then it ignores the received advertisement message; (b) If the back-off waiting timer is still valid, then it caches the received packet and waits for other possible advertisements; (c) If the received message has a better number-of-hops metric than the stored one, the latter is deleted and the former is retained. When the back-off waiting time expires, the sensor node increases the number-of-hops parameter in the retained packet and broadcasts it to its neighbours. The node will remember the address of the sender, the cluster-head, and the node from which it received the message as the nearest neighbour to the cluster-head. If a sensor node receives multiple copies of the same advertisement, it selects the one(s) with minimum number-of-hops parameter. Then the node uses the available energy to calculate a value that represents its desire to be a cluster-head in the next cluster set-up round. This value is included in the registration packet that the node sends back to the chosen cluster-head. The cluster-head will extract the highest value(s) and adds its corresponding sender to the cluster-head backup list and registers the node as a

member of the cluster. Compacting different functions into a single multi-purpose message reduces set-up communication overhead and thus makes the protocol more stable and energy efficient.

When the cluster round time is over, the current cluster-head hands the master role to the first node in the backup nodes list. With a single flood to cluster members only, the new cluster-head continues its predecessors' role without the need of further communications. The cluster-head role will also be handed to the backup node when a fault occurs in the current cluster-head node. However, in the study presented here, a limited set of faults were considered for the "hand-over" such as internal errors and the energy level approaches a threshold. In the case of faults, such as physical damage or fatal internal errors in the cluster-head, the nodes will use an alternative path until a backup node starts the process of creating a new cluster.

B. Data Transmission Phase

During the data transmission phase, sensing nodes transmit data to their cluster-head. The cluster-heads aggregate the received data before transmission to the sink or immediately multiplex messages over multiple lines in time critical applications. Each member node transmits data on its assigned time slot scheduled by TDMA. Furthermore, each cluster communicates using unique Carrier Sense Multiple Access (CDMA) codes to avoid interference with traffic generated by other clusters.

VI. SIMULATION EXPERIMENTS

Using the SenSorPlus framework, we implemented the proposed algorithm. A demonstration of the simulation is available online ¹. For our simulation, we gave all the nodes an initial supply of energy and ran the protocol until it converged. We consider the energy-efficiency of our routing protocol in terms of the number of set-up messages. For our experiments, we created a 100-node network, where the nodes are scattered randomly on 600×600 grid, such that no two nodes share the same location. Figure 1 shows a random node distribution topology of 100-nodes, where the arrows represent communicating neighbours. The power of the sensors radio transmitter is set to cover all nodes within a 20m radius. The processing delay for transmitting a message is randomly chosen between 5s and 10s. Using this network configuration, we ran the protocol and tracked its progress in terms of number of messages sent and delays. The simulation results are presented in the following subsections.

A. Efficiency and Robustness

In this subsection, we study how introducing the number-of-hops metric and the advertisements back-off waiting time can affect the energy efficient cluster formation. Figure 2 shows four network topologies each resulting from a different simulation run. In each topology, nodes are organized into

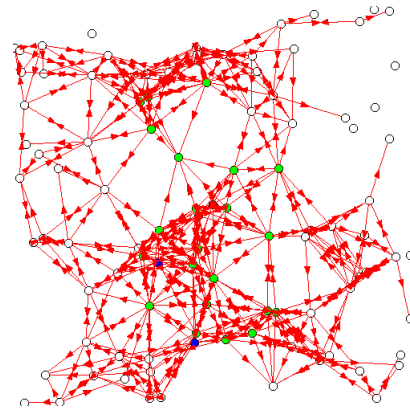


Fig. 1. A 100-nodes WSN with random topology

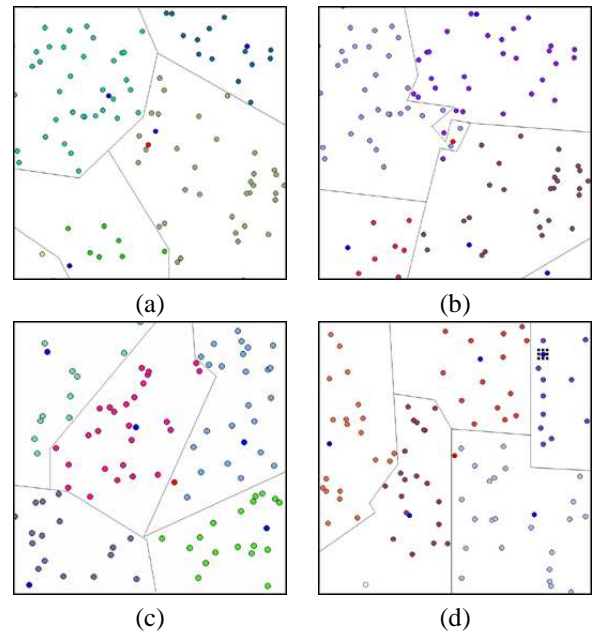


Fig. 2. Geographically uniform cluster formation

five clusters ($P=5$). The lines indicate the borders of different clusters. In each experiment we studied how the back-off waiting time and the number-of-hops metric can help to form more geographically uniform clusters.

Topologies (a) and (b) were generated using LEACH (the back-off waiting time and the number-of-hops were set to zero), whereas (c) and (d) were generated using MuMHR with the back-off waiting time set to 20s and the number-of-hops initially set to zero. Figure 3 shows the node distribution among clusters in the four network topologies. The graphs compare the distribution of nodes among clusters formed using LEACH with those formed using MuMHR. In the LEACH topology (a), the first and fifth clusters hosted over 65% of the total number of nodes while the other three clusters hosted less than 35% of the total network population. In the second LEACH topology (b), the fifth cluster had zero nodes while the percentage of nodes fluctuated among the

¹<http://www.cogentcomputing.org/cds/distributing/who/sensor/hi-cluster-routing.html>

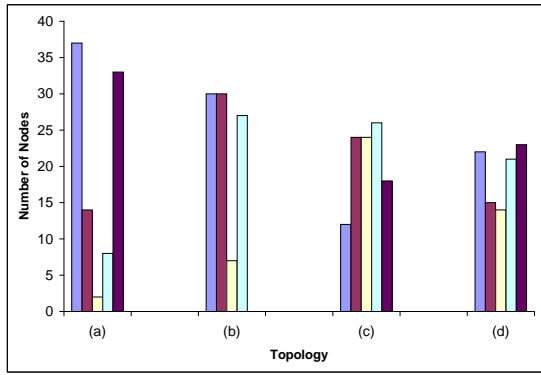
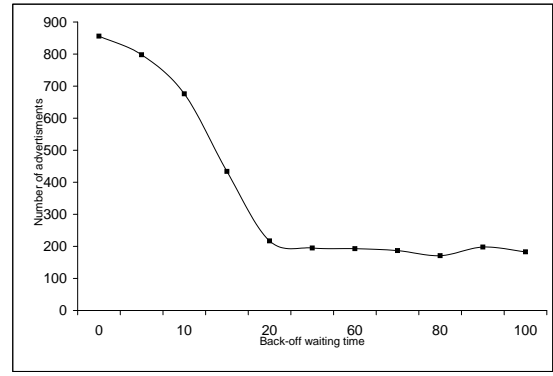


Fig. 3. Node distribution among clusters

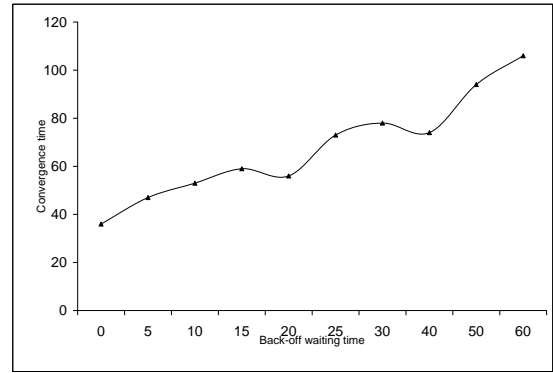
other clusters between 7% and 30%. It can be clearly seen that there is no uniform distribution of node numbers amongst the clusters, which increases both the heavy clusters management overhead and also the energy consumption. Whereas in MuMHR topologies (c) and (d), nodes were distributed much more fairly among clusters. The number of nodes at every cluster maintained a maximum of 7% difference from the optimal population (20%). In topologies (a) and (b), the area covered by different clusters varies largely. These cluster topologies are not energy efficient because data needs to traverse large number of nodes to reach the cluster-head. In MuMHR generated topologies (c) and (d), the area of all clusters is almost equivalent and nodes are distributed much more fairly among the five clusters than in (a) and (b) as figure 2 shows. This demonstrates that the back-off waiting time together with the number-of-hops metric lead to the formation of more energy efficient clusters by shortening the routes. The back-off waiting time gives more time to receive a smaller number-of-hops value. This allows nodes to register with the closest cluster-head resulting in more geographically uniform clusters. Furthermore, the number of advertisement messages is reduced, because nodes only forward the best advertisement received during the back-off waiting time. This stops unnecessary flooding at the border of neighbouring clusters.

In figure 4(a), the number of network setup messages versus the back-off waiting time is drawn. When the back-off waiting time is zero the total number of sent messages will be similar to that in LEACH. The figure shows that as the back-off waiting time becomes larger, the number of messages will decrease until the time becomes large enough to receive advertisements from all cluster-heads. It reduces the total number of set-up messages by up to 65% over LEACH. Therefore, the back-off waiting time is effective in reducing overall set-up energy consumption.

Figure 4(b), shows the network convergence time versus the back-off waiting time. A linear correspondence between the time needed to establish routes and the back-off waiting time is evident. As the back-off waiting time increases, network convergence time will increase proportionally. In time critical applications it is important that the convergence time is



(a)



(b)

Fig. 4. (a) The number of sent messages versus the back-off waiting time. (b) The convergence time versus the back-off waiting time

reduced. This means, reducing the back-off waiting time to a minimal value to capture the advantages of the back-off waiting time with minimal delay to achieve efficiency.

B. Fault-Tolerance and Reliability

Many of the proposed protocols in the field of sensor networks show poor fault-tolerance in the face of frequent node failures [9]. MuMHR provides fault tolerance through a multi-path routing strategy. The multiple paths are learned by nodes during the set-up phase through redundant messages. For example, the path to the cluster-head node is learnt from the advertisement message sent by the cluster-head. Each node joins the cluster for a certain period of time T_{nC} before it re-registers with the cluster again or registers with a new cluster. The cluster also has a life T_{C1} ; a cluster-head starts a new cluster formation round by handing the cluster-head role to the backup node. In this approach, the worst time T_R to recover from a node failure is the time taken for a node to leave a cluster or to renew registration with the same cluster-head it currently belongs to. This is written as: $T_R = T_{C1} - T_{nC}$. If all paths that a sensor node has learned fail, the node will broadcast its data to its neighbours. Neighbours will then pass the message to their cluster-head. The cluster-head may receive multiple copies of the same message and eliminates redundancy by applying aggregation. To measure the fault-

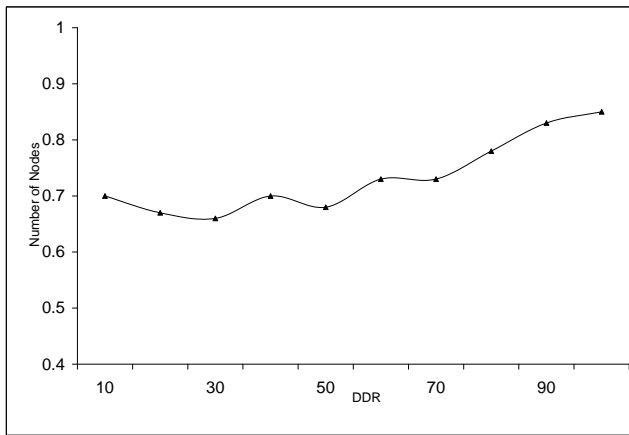


Fig. 5. DDR versus number of functioning nodes

tolerance capabilities of this protocol we make some nodes function as faulty nodes by dropping all packets that they receive, and hence affecting the communication paths. These nodes will become faulty after paths are set-up and recover to function correctly in the next cluster formation round. In the simulation, 10% of the nodes are forced to break down after paths are established. The fault tolerance capabilities of this routing protocol are evaluated using the Data Delivery Ratio (DDR) as a metric which measures the ability of the network to deliver packets to the sink through multi-paths. This measure is easy to obtain and free with every received packet. DDR is a service level parameter that indicates the network effectiveness in transmitting offered data in one direction of virtual connection [1]. It represents the ratio of packets successfully received to packet transmission attempts. Attempted packets transmitted are referred to as DataOffered. Successfully delivered packets are referred to as DataDelivered. Then the ratio can be written as: $DDR = \text{DataDelivered}/\text{DataOffered}$.

Figure 5 shows the DDR versus the number of functioning nodes. With the same configuration described in Section VI, the DDR value for ten runs each with a different network-density were drawn. The results show that the protocol maintained an average delivery ratio of 0.733. This demonstrates that multi-path routing can be used to recover from path failures and results in a better delivery ratio. DDR increases slightly as the network-density increases since higher network density slightly counteracts the effect of dead nodes. This algorithm increases the DDR and reduces energy consumption at the same time since packet multiplexing over duplicate paths helps in load balancing and prevents energy depletion.

VII. CONCLUSION

In this paper, we demonstrated MuMHR, which is an improvement over LEACH. MuMHR provides solutions to some of the limitations of LEACH. The number-of-hops parameter and the back-off waiting time resulted in more energy efficient cluster formation. The algorithm uses redundant messages received from different sources to build a multi-path map, which allows auto-adaptation to path failures. MuMHR also

enable multi-hop transmissions to relax LEACH's inflexible assumption that all nodes in the network can communicate with each other. The new algorithm achieves robustness and efficiency without location information and with less energy expenditure than LEACH. Simulation results confirm the efficiency of the algorithm in terms of communication reduction, robustness and energy savings. This routing algorithm was implemented and used by Shuttleworth et al. [13] and found to easily support various computationally demanding applications for WSNs.

ACKNOWLEDGMENTS

The authors would like to thank Sarah Mount for her help and support in using SenSorPlus.

REFERENCES

- [1] J. Dunn and C. Martin. Terminology for frame relay benchmarking. In *Internet informational RFC 3133*, June 2001.
- [2] Gumstix.com. Gumstix way small computing, 2007. [Online; accessed 26-March-2007].
- [3] M. Hammoudeh. Robust and energy efficient routing in wireless sensor networks. Master's thesis, University of Leicester, 2006.
- [4] W. Heinzelman. Application-specific protocol architectures for wireless networks. In *PhD thesis, Massachusetts institute of technology*, June 2000.
- [5] W. Heinzelman, A. Chandrakasan, and H. Balakrishna. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd International Conference on System Sciences*, January 2000.
- [6] P. Kulkarni, D. Ganesan, and Prashant Shenoy. The case for multi-tier camera sensor networks. *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 229–238, 2005.
- [7] S.H. Lee and T.C. Chung. Data aggregation for wireless sensor networks using self-organizing map. In *Artificial Intelligence and Simulation*, volume 3397/2005, pages 508–517. Springer Berlin / Heidelberg, 2004.
- [8] S. Lindsey, C.S. Raghavendra, and K. Sivalingam. Data gathering in sensor networks using the energy*delay metric. In *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, April 2001.
- [9] Y. Liu and W.K.G. Seah. A priority-based multi-path routing protocol for sensor networks. *Personal, Indoor and Mobile Radio Communications*, 1:216–220, Sept. 2004.
- [10] A. Manjeshwar and D.P. Agrawal. A protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.
- [11] A. Manjeshwar and D.P. Agrawal. Apteem: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, April 2002.
- [12] S. Mount, R. Newman, and E. Gaura. A simulation tool for system services in ad-hoc wireless sensor networks. *NSTI Nanotechnology Conference and Trade Show*, 3:423, May 2005.
- [13] J. Shuttleworth, M. Hammoudeh, E. Gaura, and R. Newman. Experimental applications of mapping services in wireless sensor networks. In *Fourth International Conference on Networked Sensing Systems*, June 2007.
- [14] S.Lindsey and C.S. Raghavendra. Pegasis: Power efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*, March 2002.
- [15] G. Smaragdakis, I. Matta, and A. Bestavros. Sep: A stable election protocol for clustered heterogeneous wireless sensor networks. In *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, August 2004.
- [16] L. Xia, X. Chen, and X. Guan. A new gradient-based routing protocol in wireless sensor networks. *Springer-Verlag Berlin Heidelberg*, 2005.
- [17] M. Ye, C. Li, G. Chen, and J. Wu. Eecs: An energy efficient clustering scheme in wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 535–540, April 2005.