EDAP: An Efficient Data-Gathering Protocol for Wireless Sensor Networks

Arash Nasiri Eghbali, Nastooh Taheri Javan*
and Mehdi Dehghan

Ad Hoc Network laboratory,
Computer Engineering and Information Technology Department,
Amirkabir University of Technology,
(Tehran Polytechnic), Tehran, Iran
E-mail: nasiri@aut.ac.ir
E-mail: nastooh@aut.ac.ir
E-mail: dehghan@aut.ac.ir
*Corresponding author

Abstract: Directed Diffusion (DD) uses data aggregation to suppress data overhead however there is no guarantee that paths from nearby sources join after a few hops. In this paper, an efficient data-gathering protocol (EDAP) is proposed to address this problem by using a Virtual Sink (VS) node nearby the sources, which plays the role of sink node and broadcasts local interest messages and routes gathered data toward destination. Also, multiple paths are constructed between VS and the sink node, which leads to load-balancing, and increase in the lifetime of the network. Simulation results show that in EDAP a significant amount of energy can be saved and the network lifetime will be increased considerably.

Keywords: WSNs; wireless sensor networks; multi-path routing; on-demand clustering; load-balancing; energy efficiency; DD; directed diffusion.

Reference to this paper should be made as follows: Eghbali, A.N., Javan, N.T. and Dehghan, M. (2011) 'EDAP: An Efficient Data-Gathering Protocol for Wireless Sensor Networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 7, No. 1, pp.12–24.

Biographical notes: Arash Nasiri Eghbali obtained his Bachelor's Degree in Electrical Engineering from Tabriz University and the Master's Degree in Computer Networks from Amirkabir University of Technology (Tehran Polytechnic) in 2007. He has been involved in research activities including design and implementation of routing, data-gathering and time synchronisation protocols in wireless sensor networks since 2004. His research interests include network architecture and protocols, distributed computing, mobile ad-hoc networks and wireless sensor networks.

Nastooh Taheri Javan obtained his Bachelor's Degree in Computer/Software Engineering from Azad University and the Master's Degree in Computer Architecture Engineering from Amirkabir University of Technology (Tehran Polytechnic) in 2007. He has been involved in research activities including design routing protocols, security and fault tolerance in mobile ad-hoc networks and data-gathering, energy consumption and routing protocols in wireless sensor networks since 2004. His research interests include network architecture, network security, mobile ad-hoc networks and wireless sensor networks.

Mehdi Dehghan is an Associate Professor in the Department of Computer Engineering at the Amirkabir University of Technology in Tehran. He received his BSc in Computer Engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 1992, and his MSc and PhD from Amirkabir University of Technology (AUT), Tehran, Iran, in 1995 and 2001, respectively. Since 1995, he has been a research scientist at Iran Telecommunication Research Center (ITRC) working in the area of Quality of Service provisioning and Network Management. He joined Computer Engineering Department of Amirkabir University of Technology in 2004. His research interests include high-speed networks, network management, mobile ad-hoc networks and fault-tolerant computing.

1 Introduction

Wireless Sensor Networks (WSNs) consist of a number of sensor nodes, scattered in the environment to sense special events such as searching a mobile target or measuring the amount of radioactive radiation in a specified area. These nodes are usually identical with a limited amount of energy. Instead of unique addresses, they are identified by the information gathered by them. They are mostly implemented densely and thus a number of nodes are triggered by a single event. In such situations, each node starts to send the gathered information towards the sink, which is usually located far from senders.

DD is a data-centric routing protocol that uses only local interaction between neighbour nodes (Intanagonwiwat et al., 2002). In DD, attribute-value pairs are used for describing information and data. This algorithm in its basic form has two phases. In the first phase, the sink nodes flood a request packet called 'interest', which consists of the desired attribute-value pairs. When this packet reaches a source node that has the requested information (second phase), it floods an 'ED' packet through the network. When this packet reaches the sink node, it will send a 'positive reinforcement' packet towards source node. This packet is being forwarded through the path traversed by the ED packet. In this way, a bi-directional path is constructed between two nodes. Afterwards, data packets will be sent through reinforced paths. This algorithm is also called Two-Phase-Pull (TPP) algorithm (Heidemann et al., 2003). Also by assuming the connections to be bi-directional (which in most cases is not true), One-Phase-Pull (OPP) algorithm can be used. In this approach, data packets are sent immediately after the interest packet reaches the source node. Therefore, in OOP algorithm, the cost of ED packet flooding will be omitted.

To provide connection between sinks and sources, this work relies on low-rate flooding of events, enabling local re-routing whenever the nodes in the primary path have failed owing to energy consumption. In sensor networks, where energy efficiency is of paramount importance, such flooding can adversely impact the lifetime of network.

In DD, data aggregation and in-network processing approaches have been introduced to suppress the additional data overhead. In this protocol, the paths from the adjacent sources often join together after a few hops but there are no guarantee for such combination. In cases where the sensed event is spread geographically, the probability of such combination is reduced. Another problem of DD, which arises in the presence of many source nodes near a single event, is the mechanism used in this protocol for path construction between sources and sinks. By receiving an interest, each source floods an ED. When this packet reaches the destination, the path traversed is reinforced by the sink and used for later data transmissions. The procedure is repeated for each source separately and, thus, a significant amount of network energy is consumed.

The EDAP is proposed to address the problems of late-aggregation and separate ED-flooding. In our local clustering protocol, early aggregation can be achieved by using a VS near the sources, which plays the role of sink node and broadcasts local interest messages. Therefore, data packets are sent preliminary towards VS node. VS node undertakes the responsibility of sending the data packets towards destination. Hence, there is no need for different sources in a single cluster to broadcast separate ED messages. This operation is done only once by the VS node. So, the routing protocol overhead is reduced significantly using our clustering method.

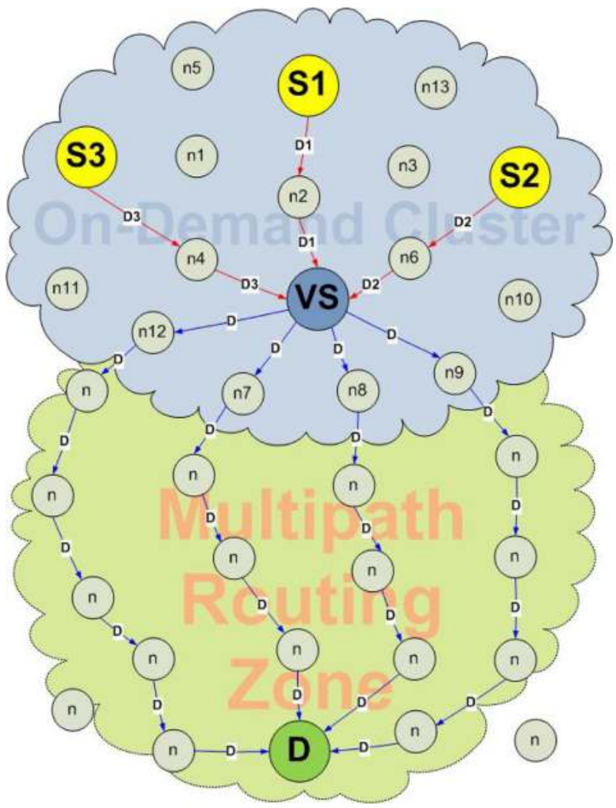
Another problem is the mechanism used in DD for routing selection, which mostly leads to the selection of the shortest path between sinks and sources. In this case, the nodes in the shortest path will fail after a short period of time due to lack of energy. This problem is intensified when the nearest route to the shortest path is selected after path refresh period, which is quite probable. In this situation, network partitioning will occur along the depleted paths.

To avoid this, in our paper, we propose using multi-path routing algorithms to increase load-balancing between network nodes during forwarding data packets, between VS and sink nodes. In this way, we can reduce the rate of interest packet flooding by increasing the lifetime of connection. We consider a proactive approach to construct multiple paths between two nodes and use node-disjoint multi-paths where the alternate paths do not intersect each other. The disjoint property has better performance for our load-balancing technique but reduces the number of paths, created by the algorithm. Also, if a single node in a disjoint path fails, other nodes will be left unused until next refreshment period. In sensor networks, designing such algorithms has been proved to be a difficult task, owing to their data-centric routing with localised path set-up as indicated in Ganesan et al. (2001).

Limited Forward Improvement (LFI) method is used in this paper to improve the probability of construction of multiple paths. We introduce the premier packet problem where the first flooded packet dominates most of the nodes between source and destination and prevents our proactive algorithm to construct multiple paths. Using LFI, each node would selectively send the ED to nodes nearer to destination instead of broadcasting them. In this way, we can achieve a significant reduction in the overhead imposed by DD to broadcast the ED packets all through the network. The number of constructed paths would also grow using this improvement method. This method can be used instead of TPP DD, which uses a network wide flooding for each source node and OPP DD where all the connections are assumed to be bi-directional.

Each of these algorithms separately has its own weak points. When using on-demand clustering protocol, all paths from different sources will merge after a few hops and all of them are forwarded by VS node towards the sink, using a single path. As mentioned earlier, this may cause the network to be partitioned. Also, multi-path routing algorithm constructs disjoint paths between sources and sink nodes and prevents aggregation. The Efficient Data-Gathering Protocol (EDAP) uses both algorithms to gain the benefits of each one and bypass their defects. A sample of routing scheme in EDAP is shown in Figure 1.

Figure 1 Sample EDAP routing scheme (see online version for colours)



The rest of this paper is presented as follows: in Section 2, we will introduce On-Demand Clustering Protocol (ODCP) and describe the algorithm in detail. In Section 3, we explain the algorithm used for multiple paths construction and also introduce LFI method. We also present our load-balancing mechanism used after multi-path construction phase in this section. In Section 4, we will explain the EDAP algorithm and discuss about the problems that we may face during the combination of the on-demand clustering and multi-path routing algorithms. The methodology we used for implementing and testing these protocols and simulation results are available in Section 5 and a comparison between the original DD algorithm and our proposed algorithms will be given in this section. Related works are reviewed in Section 6 and finally we will conclude the paper and present future works to improve our routing algorithm in Section 7.

2 On-Demand Clustering Protocol

ODCP is used for collecting information gathered by nearby sensor nodes (Nasiri Eghbali et al., 2007). Then, these data are aggregated and sent towards sink by the cluster head named as VS. In this section, the ODCP will be introduced and all the mechanisms used for cluster head selection and routing within a cluster will be described in detail.

This protocol has four phases. First is the VS selection in which a suitable cluster head is selected among nodes near the sources. In the second phase, selected VS will explore a path to the sink, and in the third phase, after a period of time, a new VS is selected among neighbours of prior VS. The fourth phase is considered for situations where the sink node is crashed or lacks enough energy to continue forwarding data packets towards the sink.

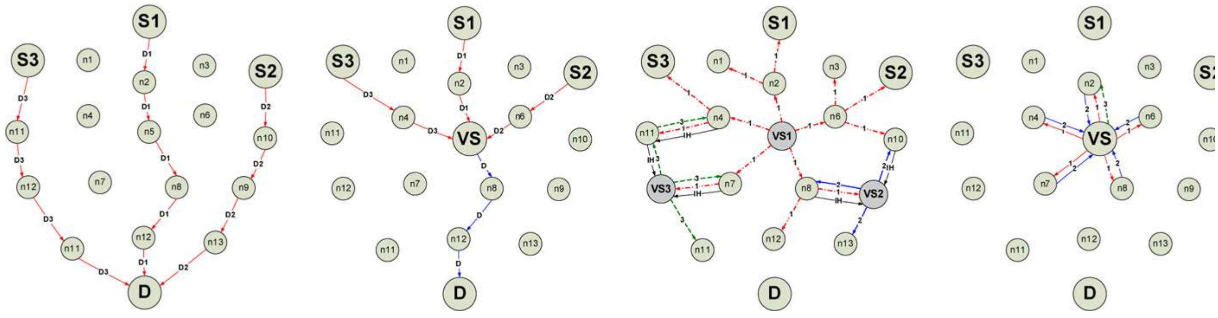
Phase I: Virtual Sink Selection – VS selection is the most important and challenging phase in this protocol. As DD is a localised algorithm and each node has only local information, selecting a single node as VS node between others seems to be a difficult task. The first idea, which crosses the mind, is to select one of the sources as VS. Problems will occur when sources are not in radio range of each other. Hence, in such case, multiple VSs will be selected and we need to choose one of them as VS. In ODPC, one of the nodes in original path between source and sink will be selected as VS. Distance of this node from the source (Dsrc) can be selected, considering density of nodes and spatial properties of sensed event. As we usually prefer to increase the lifetime of source nodes, this selection seems to be a better one. In our simulations, Dsrc assumed to be at most 2 hops. When a node is selected as cluster head, it broadcasts an interest message to all nodes in the cluster. For this purpose, limited flooding method is used. The interest packets are tagged by VS. This tag indicates the Time-To-Live (TTL) of forwarded message and each packet will be ignored after traversing such amount of nodes. The TTL value is decremented in each hop until it becomes zero. Therefore, the overhead of such flooding is directly related to the size of cluster, which is very small in comparison with network size.

In this step, the number of selected VS nodes is the same as number of sources. This selection is shown in Figure 2(c). First, VS1, VS2 and VS3 are selected as VS; so, one of these VS nodes should be selected as the final VS. A simple approach for such selection is tagging the VS interest messages by VS selection time. Cluster nodes (including VS nodes) can select VS with the least time-stamp. So, the nearest VS to the sink is selected. Overhead of such flooding increases linearly by the number of sources, which is not acceptable especially in high-density networks.

In Figure 2(c), the VS selection mechanism has been shown. Here, the VS1 node is the VS with the lowest time-stamp. So, the packets from VS2 and VS3 are not flooded in cluster. The paths used by the original DD are depicted in Figure 2(a) and the paths used by ODCP are shown in Figure 2(b). In ODCP, we used TPP algorithm to route packets within the cluster and sources send local ED to the VS and it reinforces them.

Using this method, the nodes having a distance less than Dsrc from the selected VS node will form a cluster.

Figure 2 The on-demand clustering protocol: (a) sample routing scheme for directed diffusion; (b) sample routing scheme for clustered directed diffusion; (c) virtual slink selection during first round and (d) virtual sink selection during next rounds (see online version for colours)



(a) (b) (c) (d)

Phase II: Route Discovery – After VS selection phase, we should construct a path from each source towards VS and a path from VS to the sink. Two approaches are used for path construction within the cluster. First is using OPP algorithm within the cluster. In this case, sources will send data through the path traversed by local interest messages. As we did not assume the links to be bi-directional, the realistic approach is using TPP algorithm in which sources will broadcast a local ED packet in the cluster and VS node reinforces the constructed path after receiving an ED packet from each source.

Then, to find a path towards the sink, TPP algorithm is used and VS broadcasts an ED packet. When this packet reaches sink, this node will reinforce the path traversed by the ED packet and a path will be constructed from VS towards the sink.

Phase III: Virtual Sink Refreshment – All the traffic generated in the cluster is passed through the VS node. Therefore, the energy of this node will be consumed after a short period of time. So, to avoid VS failure and to perform load-balancing, after the VS refresh period, VS selects one of its neighbours as the next VS. At first, the VS will broadcast a neighbour request message. Then, it waits for a short period of time. During this period, VS neighbours will receive this message and send a neighbour reply message, which contains energy of each neighbour. So, VS can select the neighbour with the largest amount of remaining energy. Then, it will send a new sink selection message to the selected node. Other important parameters for selecting a node as VS can be its distance from the sink node and other sources but considering complexity of the protocol needed for evaluation of these parameters, it is not a realistic choice. The new VS selection procedure is shown in Figure 2(d). In this figure, the neighbour request messages are labelled as '1', the neighbour reply messages as '2' and the new sink selection message as '3'.

If a new VS cannot find a path towards the sink after a period of time called PATH_EXPLORATION_PERIOD, it will select another neighbour and the VS refreshment steps will be repeated.

Phase IV: Virtual Sink Expiration – There are two cases in which the VS is expired. One is when it does not receive any data packet for a period of time called EXPIRATION_PERIOD. The second case is when the remaining energy of VS and all its neighbours become below a threshold called VS_MINIMUM_ENERGY. In this case, the VS node will broadcast a VS_DISABLE message in the cluster. The sources after receiving this message will broadcast an ED towards the sink and send the data directly to the sink node. Additionally, each node has an expiration timer. This timer is rescheduled by receiving each VS interest message. So, if the VS node fails, sources will themselves take the responsibility of sending data towards the sink.

3 Multi-path construction methods

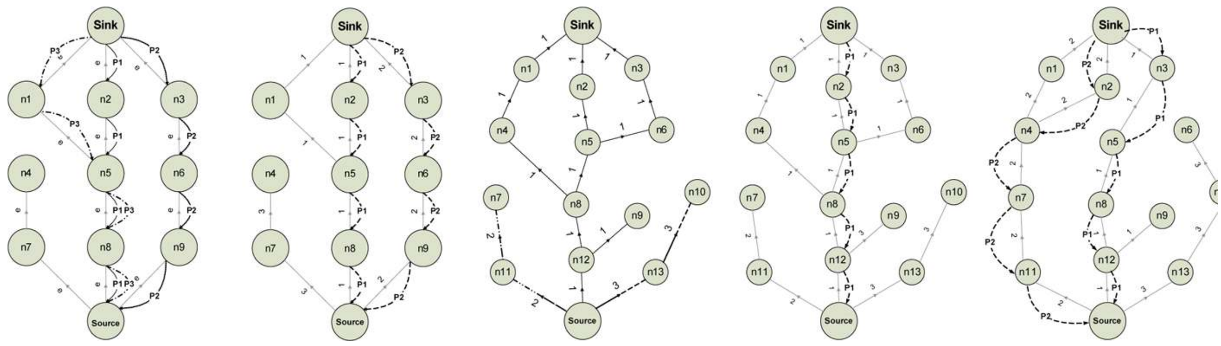
In this section, we introduce our multi-path construction methods (Nasiri Eghbali and Dehghan, 2007). Also, we would have a brief evaluation and comparison of these methods.

3.1 Simple multi-path routing method

Unlike DD in which the sink node only reinforces the first arriving ED packet and simply drops others, in this method the sink node reinforces all the received ED packets. Also, when an old positive reinforcement packet arrives in a node in the path (path node), instead of just dropping the old reinforcement packet (as in DD), it will send a negative reinforcement message to its previous hop. This is done to avoid forming none-disjoint paths between sources and sinks. This is used in Ganesan et al. (2001) to increase the resiliency of DD routing protocol.

As another approach, the path node can also forward the old positive reinforcement message, through the path already traversed by the prior reinforcement message. This way, paths may not be completely disjoint but the number of constructed paths will be fairly increased. These methods are explained in Figure 3. The ED that first arrived in each node is labelled as '*e*'. In Figure 3(a), we have depicted the disjoint path creation method. As it is shown, in this method node n5 sends a negative reinforcement message to node n1 after receiving the old P3 reinforcement message.

Figure 3 Comparison between simple and proactive multi-path construction algorithms: (a) simple shared multi-path construction; (b) proactive disjoint multi-path construction; (c) premier packet problem; (d) proactive disjoint multi-path directed diffusion and (e) disjoint proactive multi-path method (using LFI technique) (see online version for colours)



(a) (b) (c) (d) (e)

3.2 Disjoint proactive multi-path construction method

This method produces multiple paths proactively by tagging the ED messages in the source node, before forwarding them to the neighbour nodes. We call this tag, multi-path identifier as MP_ID. Basically, this approach acts similarly as the simple disjoint path construction method but when an ED arrives in the sink node, it will be recognised as a new ED with a new MP_ID or a duplicated one. So, the sink reinforces only the first ED packet among those arriving with the same MP_IDs. In this way, we will be assured that created paths are disjoint (the old ED packets are ignored in the base DD algorithm).

This method is shown in Figure 3(b) in which the sink node receives the ED packets with MP_IDs 1 and 2. First, the sink receives the ED from node n2 with MP_ID, then receives the ED from node n3 and finally receives the ED from n1. So, the sink reinforces only the gradients from nodes n2 and n3 and ignores the ED received from n1.

The simple and proactive multi-path routing algorithms are based on the arrival of ED packets in the sink node with totally distinct path nodes. Unfortunately, this rarely happens because of the broadcasting and physical nature of wireless medium, which reduces the probability of arriving multiple different ED packets. The first packet arrives at the sink node, usually traverses all the neighbour nodes and the other ED packets can hardly reach them.

3.3 Limited Forward Improvement method

The functionality of simple multi-path construction methods mostly is related to the topology of the network. The main weakness of these methods relies upon this fact that it rarely can produce multiple disjoint paths. Disjoint routes, starting from the source node, will merge and become a single path after traversing just a few nodes. The main reason behind this problem is the nature of wireless medium and contentions occur during flooding a packet in a wireless network. When the sink node broadcasts an ED packet to its neighbours and they forward the packets to their neighbours, the wireless medium will soon become saturated. In this case, the first ED packet that is sent towards the source node (out of the saturated area) can traverse most of the nodes between the source and the sink nodes. We call this event the 'premier packet problem', which reduces the probability of arriving multiple distinct ED packets in the sink node. The first packet arrives at the sink node, usually traverses all the neighbour nodes and prevents other ED packets to reach the sink node.

The DD algorithm makes use of this event for data aggregation but this event prevents the simple and proactive routing methods to make multiple disjoint paths. This event has been depicted in Figure 3(c). In this figure, the ED packet with MP_ID 1 has reached the sink before others.

LFI method is introduced to solve this problem. In this method, each node selectively forwards the ED packets to first F nearer nodes to the sink. The distance is measured by the time-stamp, saved during the interest message broadcasting (in the first phase of the TPP algorithm). An example is shown in Figure 3(d).

The LFI method leads to better results and can usually create a notable number of disjoint paths. Another advantage of using this method is reducing the cost of flooding ED packets. Because of its limited forwarding, the number of flooded ED packets will be increased linearly by the size or density of the network. Although in this method the nodes should send the ED packets to each neighbour instead of broadcasting, this method can reduce the total number of communication overhead. In Figure 3(e), the result of using LFI method has been depicted.

3.4 Braided proactive multi-path construction method

For increasing the number of constructed paths, each node can forward more than just a single ED packet. In this case, the duplicated ED packets with different MP_IDs are not discarded and each node is permitted to forward at most T duplicate packets. We call this parameter, the node forwarding threshold parameter. Therefore, the paths constructed are not necessarily disjoint and can share a number of nodes. Also, in this method, gradients are not shared between two different paths and each gradient can only be a member of a single path. As the numbers of forwarded exploratory packets are increased linearly by T value, this method can only be used with LFI method and in this case the imposed routing overhead is almost negligible.

3.5 One and a half phase pull Directed Diffusion

The proposed proactive multi-path construction methods stand somewhere between OPP and TPP algorithms (Heidemann et al., 2003) and can be named as One and a Half phase Pull Directed Diffusion (OHPP) algorithm. In the OPP method, the ED packets are omitted and the source node will send data packets to the first neighbour node that has sent the interest packet. In this algorithm, we assume the connections between nodes to be bi-directional but this assumption usually is not true. The TPP DD algorithm has also the problem of large overhead, caused by flooding ED packets through the whole network. In our multi-path routing algorithm, the connections need not to be bi-directional and by using LFI techniques, the overhead of ED packets is almost negligible compared with the TPP algorithm.

3.6 Energy efficiency and load-balancing

As mentioned in the introduction, sensor nodes are energy constrained. In this section, we present methods for implementing load-balancing mechanisms using the multiple paths, created by the proposed algorithm in the prior sections.

As we know, most of the energy utilised in a sensor node is usually consumed for the routing procedure especially in large-scale sensor networks where the distance between the source of data and the destination is significant. So, it is very important to spread this overhead between different nodes. In this paper, the multi-path routing method is used mostly for load-balancing, although constructing a couple of disjoint or partially disjoint paths between sink and source will lead to longer lifetime of each path and also the connection will be more robust against failures. Therefore, the rate of interest flooding can be reduced, which is an expensive operation owing to its large communication overhead, although this subject is beyond this work. In this paper, we will focus on load-balancing between multiple paths created by the proposed multi-path routing algorithms.

A simple approach to perform load-balancing is spreading data packets uniformly between different paths. Then, PATH_ALIVE messages can be sent periodically through each path. If a path becomes unusable due to energy consumption of its nodes or other reasons, this path will not be selected anymore.

In this paper, we present a more efficient approach in which the path minimum energy (e) and the path length (l) are regarded during the packet spreading process between the alternate paths.

Path minimum energy e is defined as the energy of node with the least amount of energy between the nodes along the path, and the path length is defined as the number of hops in that path. For reinforcing the paths with bigger amounts of e, we consider the probability of sending a packet along a path, directly proportional to its minimum energy. Let $e_p(i)$ be the path minimum energy of path with MP_ID i and $P_{\text{sel}}(i)$ be the probability of selecting path i. In this case:

$$P_{\text{sel}}(i) \propto e_p(i). \tag{1}$$

Another parameter used is the path length l, as the path length grows, the cost of routing along that path would be increased linearly by the length of it. So, we prefer to select the paths with shorter lengths. So:

$$P_{\text{sel}}(i) \propto \frac{1}{l_p(i)}. \tag{2}$$

By this approach, at first, shorter paths have more chance to forward the data packets, but after a period of time, their energy would be decreased. In this situation, the longer paths with more energy would have a better chance of being selected by the source node.

To further improve this, we introduce two thresholds named as minimum energy threshold (eth) and maximum path length (lth). When the energy of a path reaches below eth, or its length is more than lth, the probability of its selection by the source node, calculated according to formulas (1) and (2), is multiplied by 0.1. In our simulations, we use the lth with value $l\text{min} \times \beta$ and the eth is selected as $e\text{max}/\alpha$ where $l$min is the length of the path with the minimum length and $e$max stands for the e (minimum energy) of the path with the maximum e (when these parameters are calculated and where are they maintained and how are they updated). So, the probability for the source node to select a path between multiple paths is calculated by:

$$P_{\text{sel}}(i) = c\frac{e(i)}{l(i)}. \tag{3}$$

And, n is the number of paths constructed by the routing algorithm and α and β are threshold factors. Here, we assumed these values to be $\alpha = 2$ and $\beta = 4$ according to our simulation experiments.

$$e(i)=\begin{cases}e_p(i) & e_p(i)>e_{th} \text{ and } n>0\\ 0.1\times e_p(i) & e_p(i)<e_{th}\end{cases},\ e_{th}=\max[e_p(i)]/\alpha$$

$$l(i)=\begin{cases}l_p(i) & l_p(i)>l_{th} \text{ and } n>0\\ 10\times l_p(i) & l_p(i)<l_{th}\end{cases},\ l_{th}=\min[l_p(i)]\times\beta \quad (4)$$

$$c=\frac{1}{\sum_i^{i=n}\frac{e(i)}{l(i)}=1}. \quad (5)$$

The *e*th changes dynamically during the routing period. To inform the source node, about the energy of each path, periodic PATH_ALIVE messages are sent along every path. The PATH_ALIVE messages that are used for path refreshment are the same as POSITIVE_ REINFORCEMENT messages and calculate the minimum energy of the paths. Using this method, the connection time between source and sink can be increased and the number of packets dropped during the routing process can be reduced.

Another way to increase the energy efficiency of routing protocol is using the LFI method that reduces the overhead of flooding ED packets. As we will see in the simulation section, this can increase lifetime of the network significantly.

4 EDAP data-gathering protocol

As described in former sections, ODCP algorithm aims to reduce the routing overhead by omitting extra ED packets, broadcasted by nearby sources. In this algorithm, nearby sources forward their data towards a node near them, which plays the role of the sink node named as VS. Then, all data will be aggregated in this node and sent through a single path towards the original sink node. Therefore, this node plays the role of single virtual source either.

As highlighted before, using a single path to forward all data gathered by sources has the problem of network partitioning. EDAP algorithm uses multi-path routing methods presented in Section 2 for implementing load-balancing between the VS and the sink node. Disjoint multi-path construction method prevents the gathered data to be aggregated but by using on-demand clustering technique, all the data sent towards the sink through multiple paths are already aggregated in the VS node. Figure 1 shows the way data are gathered using EDAP.

5 Methodology and simulation

In this section, the methodologies and assumption used for our simulations are described and the simulation results for each proposed algorithm are presented.

5.1 Methodology

Performance evaluation experiments for WSNs are faced with a number of practical and conceptual difficulties. Here, we summarise our main choices for the simulation set-up. In this paper, ODCP and Multi-path Directed Diffusion (MDD) are separately compared with the original DD algorithm.

Protocol version

We simulated DD algorithm available with ns-2 simulator version 2.30. This protocol is implemented for simulator in two versions (Heidemann et al., 2002) named diffusion and diffusion3. Our simulations are based on the diffusion3 protocol, which is a complete protocol implementation and allows a more realistic evaluation of the protocol.

Energy model

In the original DD, the IEEE 802.11 is used for the MAC layer. For comparability, we used the same MAC layer and energy model as in Intanagonwiwat et al. (2000) that is the PCM-CIA WLAN card model in ns-2. This card consumes 0.660 W when sending and 0.395 W when receiving. In this paper, the transmission range is assumed to be fixed and 200 m.

Load model

The traditional DD algorithm floods an interest message, every 30 s and ED floods every 50 s. We used these predefined values in our simulations. In these simulations, we used the ping application as the network traffic with different packet rates. In ODCP algorithm, the rates of 1 packet per second is used and for MDD two different packet rates of 1 and 10 packets per second are used and in EDAP data rate of 0.5 packets per second is used. Also, the VS expiration period is assumed to be 60 s and 120 s.

Overhead, drop percentage and delay computation

To compare delay, overhead and drop percentages, between DD and ODCP, these parameters were measured during 100 s using variable number of sources. We measured the number of none-data packets, during the connection time and divided it by the number of received data packets to compute overhead. The average delay is also calculated using ping timestamps for each routing algorithm. Drop percentage was measured by dividing number of dropped packets by total number of packets sent.

For comparison between MDD and DD overhead, we measured the number of none-data packets, during receiving 100 data packets in the sink. Our measurement may not be quite precise but it helps us to have a slight comparison between the overhead, imposed by different algorithms.

Also for measuring delay and connection lifetime, the source or sources start to send ping data packets towards the sink continually, until the connection is broken, due to path node failures caused by energy depletion. This period is measured and considered as connection lifetime. The average delay is also calculated for each routing algorithm. The initial energy of all nodes in the network assumed to be 5 joules. In our simulations, effect of changing VS refresh time using 6 sources and effect of increasing number of sources on connection lifetime was studied. We assumed the initial energy of all nodes in the network to be 5 joules.

Multi-path calculation and evaluation

In our approach, for comparison between suggested multi-path routing protocols, protocol is tested in a 10×10 grid (100 nodes), with one sink and one source. Each protocol has been simulated at least five times and the mean value of each measurement has been considered. This iteration was quite necessary, for the random behaviour of most proposed algorithms. Then, we measured the following parameters: number of paths, total number of share nodes among the paths, average minimum and maximum path length and finally mean path length. We used four densities to highlight the effect of density of nodes on the results of different path construction algorithms. To change network density, the minimum distance between two adjacent nodes in the grid is assumed to be 75, 100, 150 and 200 m.

Energy calculations for ODCP

ODCP algorithm aims to decrease the overhead of routing in network nodes by omitting extra ED packet flooding. For measurement of energy efficiency of ODCP and comparison between ODCP and DD, we used the scenarios presented in III.D. Initial energy of all network nodes is assumed to be 50 joules. In our scenarios, the sources start to send ping data packets towards the sink continually, during simulation period (250 s) and average energy of nodes was measured each 50 s.

5.2 Simulation results

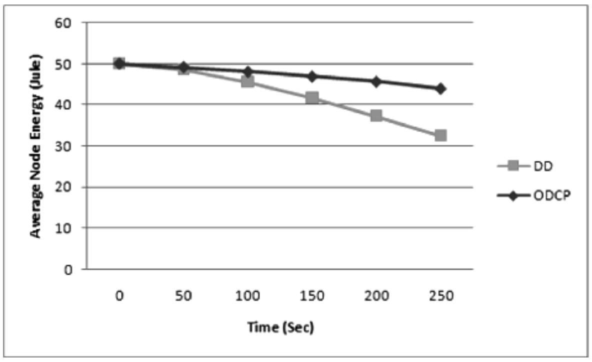
In this section, we will show the simulation results, achieved by implementing the scenarios and assumptions, described in this section for ODCP, MDD and ODCP-MP algorithms.

5.2.1 On-Demand Clustering Protocol (ODCP) simulation results

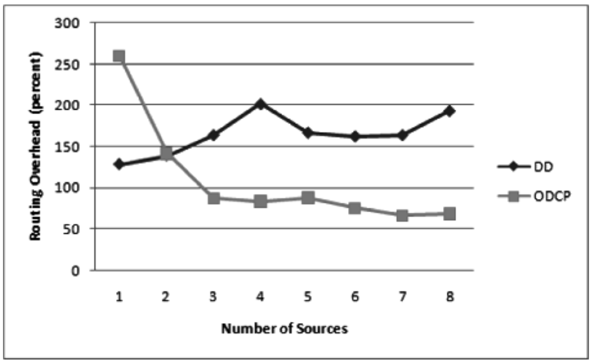
Energy efficiency

As it can be figured from Figure 4(a), the energy consumption in ODCP is less than DD. This is due to omitting the ED packets for each source node. Also, it is obvious that by increasing the number of sources or increasing network size in comparison with cluster size, the efficiency of ODCP will be increased compared with DD algorithm. Although by using ODCP algorithm, average energy of all network nodes is increased, average energy of cluster nodes will be decreased due to clustering protocol overhead.

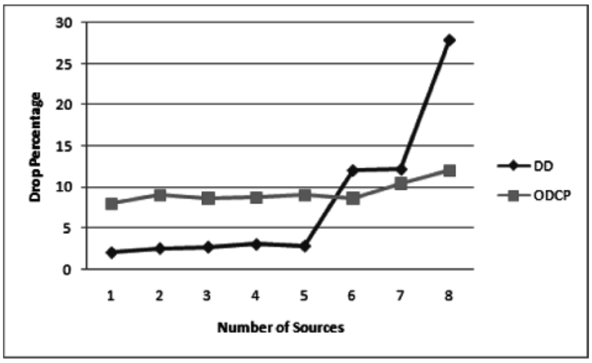
Figure 4 On-demand clustering protocol simulation results: (a) average node energy vs. time; (b) routing overhead percentage vs. number of sources; (c) drop percentage vs. number of sources; (d) connection lifetime vs. number of sources; (e) connection lifetime using different refresh periods and (f) number of delivered packets using different refresh periods



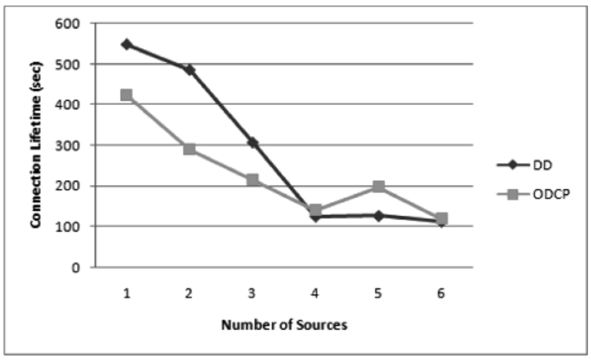
(a)



(b)

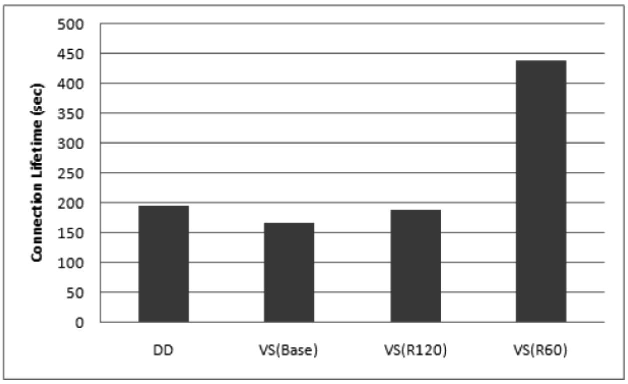


(c)

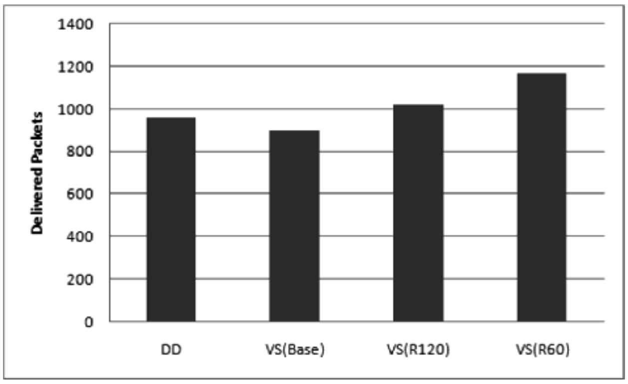


(d)

Figure 4 On-demand clustering protocol simulation results: (a) average node energy vs. time; (b) routing overhead percentage vs. number of sources; (c) drop percentage vs. number of sources; (d) connection lifetime vs. number of sources; (e) connection lifetime using different refresh periods and (f) number of delivered packets using different refresh periods (continued)



(e)



(f)

Connection lifetime

Connection lifetime has been shown in Figure 4(d). The results show that the connection lifetime will be decreased using the ODCP algorithm using less than 4 sources. This decreasing is due to clustering protocol overhead. However, lifetime will be improved when the number of sources grows. In these simulations, a fixed VS is used during connection lifetime and VS refresh phase has been omitted.

In Figure 4(e), effect of changing VS refresh period on connection lifetime has been shown. In this figure, VS(R60) and VS(R120) represent using refresh periods of 60 s and 120 s respectively and VS(base) stands for ODCP without VS refreshment phase. Result shows that by decreasing the VS refresh period, although routing overhead will be increased, connection lifetime will be increased significantly. However, as it is shown in Figure 4(e), number of delivered packet does not increase as connection lifetime. So, in this case after 200 s, most of the sources are disconnected from the sink but at least one of them is connected and therefore connection is not broken until 430 s.

Drop percentage

Drop percentage will be increased using ODCP approach because during cluster set-up, packets will be dropped (Figure 4(c)) but this value remains constant and is not sensitive to increasing number of sources but in DD by increasing number of sources drop percentage will be increased considerably.

Average delay

The delay will be decreased using our ODCP. This decreasing is due to lack of network contention produced by flooded ED. However, network response time will be decreased because of time needed for cluster set-up and VS selection procedure.

Routing overhead

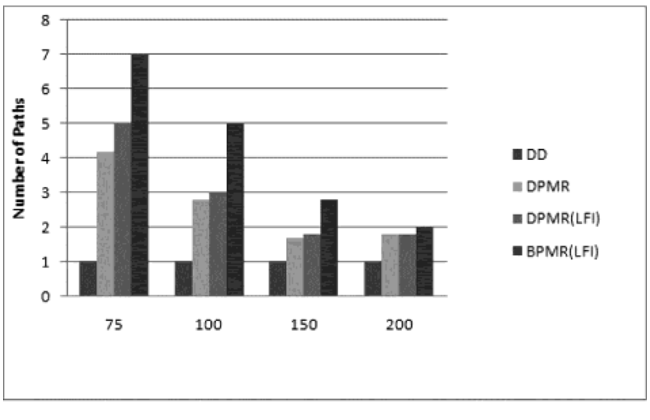
The reduction of routing overhead using ODCP compared with DD has been shown in Figure 4(b). This reduction is due to omitting extra ED packets.

5.2.2 Multi-path Directed Diffusion (MDD) simulation results

Multi-path construction

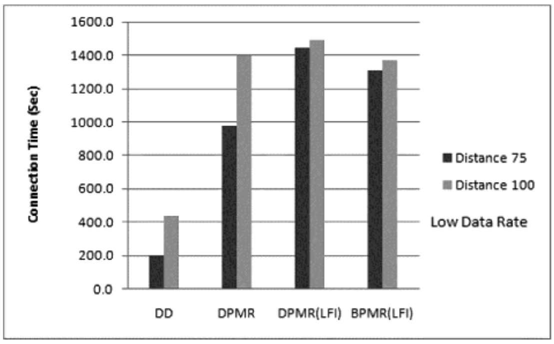
In Figure 5(a), the number of paths, constructed by different approaches, has been shown for different algorithms. Two main results can be extracted from this chart. First, the number of paths produced by the different multi-path algorithms grows by increasing the density of the network. Second, among the presented methods, braided proactive multi-path routing algorithm (using LFI method) can produce more paths than the other methods.

Figure 5 Multi-path routing simulation results: (a) number of paths vs. grid distance; (b) connection lifetime for different multi-path routing protocols with low data rate; (c) connection lifetime for different multi-path routing protocols with high data rate; (d) number of none-data packets for different multi-path routing protocols; (e) delay for different multi-path routing protocols with low data rate and (f) drop percentage for different multi-path routing protocols with low data rate

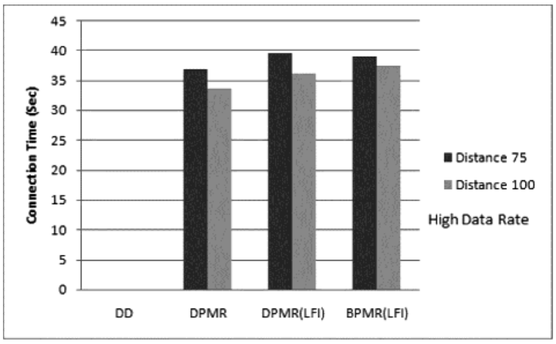


(a)

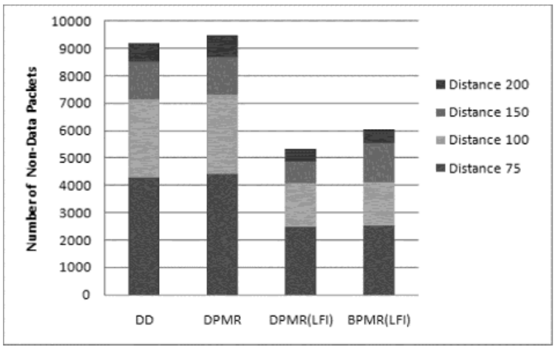
Figure 5 Multi-path routing simulation results: (a) number of paths vs. grid distance; (b) connection lifetime for different multi-path routing protocols with low data rate; (c) connection lifetime for different multi-path routing protocols with high data rate; (d) number of none-data packets for different multi-path routing protocols; (e) delay for different multi-path routing protocols with low data rate and (f) drop percentage for different multi-path routing protocols with low data rate (continued)



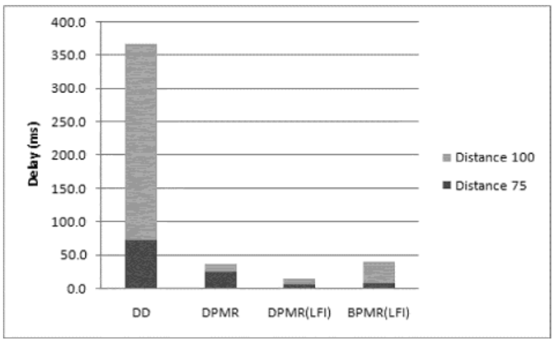
(b)



(c)

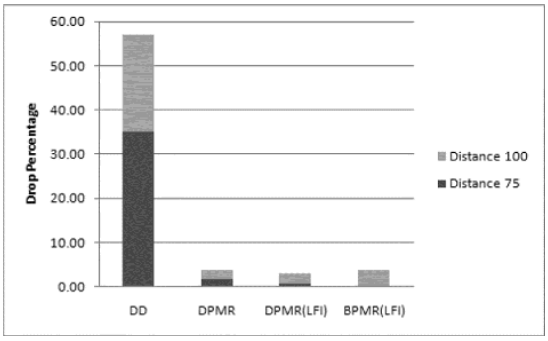


(d)



(e)

Figure 5 Multi-path routing simulation results: (a) number of paths vs. grid distance; (b) connection lifetime for different multi-path routing protocols with low data rate; (c) connection lifetime for different multi-path routing protocols with high data rate; (d) number of none-data packets for different multi-path routing protocols; (e) delay for different multi-path routing protocols with low data rate and (f) drop percentage for different multi-path routing protocols with low data rate (continued)



(f)

Connection time

Using multi-path routing and load-balancing approaches will increase the connection time between source and sink. As shown in Figure 5(b) and 5(c), multi-path routing has a prominent effect of the connection lifetime. The proactive multi-path routing algorithm performs much better than the original DD algorithm. This is due to the limited forwarding improvement of this approach, which reduced the overhead of ED packet flooding. Also, in high data rates (10 packet per second) where the original DD fails, MDD algorithm can deliver packets to destination.

Overhead estimation

The overhead of the DD will decrease by using LFI method and the proactive routing algorithms show a significant reduction in routing protocol overhead in comparison with DD algorithm (Figure 5(d)).

Delay computation and drop percentage

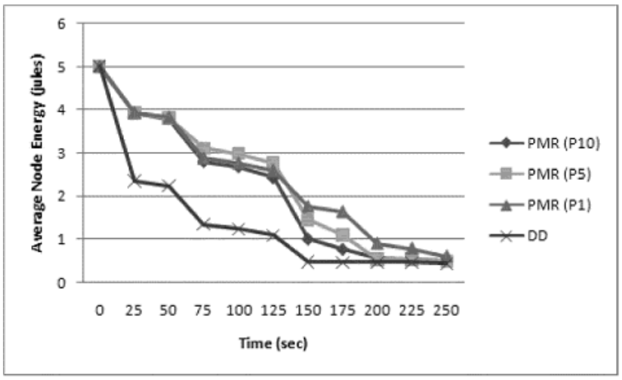
As it is shown in Figure 5(e), the multi-path routing methods will reduce the delay significantly due to omitting the extra ED packets. The drop percentage is also improved significantly using multi-path methods as depicted in Figure 5(f).

5.2.3 EDAP simulation results

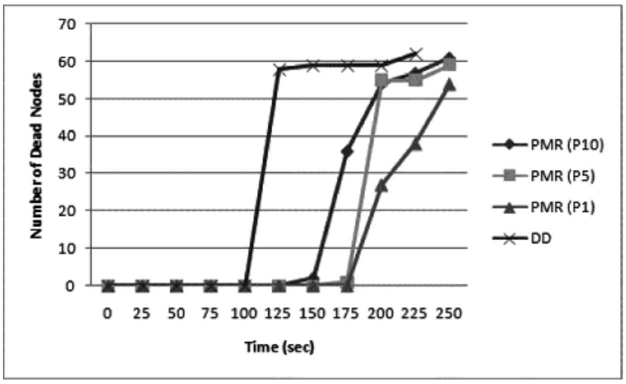
Energy consumption

As it is shown in Figure 6(a), using EDAP will increase the average energy of network nodes but due to overhead of multi-path construction methods, by increasing the number of paths between VS and sink nodes in EDAP, average energy of network nodes will be decreased but better load-balancing will be implemented. In this simulation, 4 sources with different numbers of paths are used between VS and the sink node.

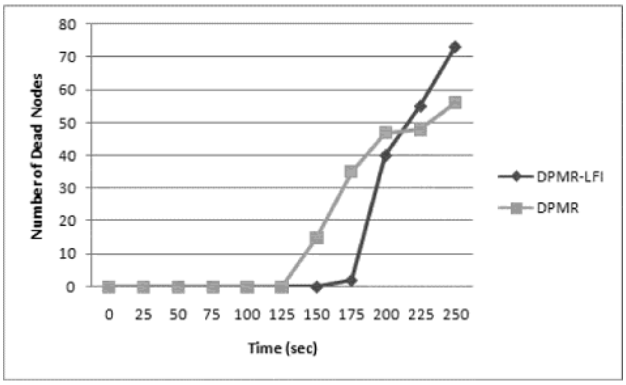
Figure 6 EDAP routing simulation results: (a) energy usage comparison between DD and DPMR routing method with different number of paths using 6 sources; (b) network lifetime comparison between DD and DPMR routing method with different number of paths using 6 sources; (c) network lifetime comparison between DPMR and improved DPMR multi-path routing methods



(a)



(b)



(c)

Network lifetime

If we define the network lifetime by the period in which at least 50% of network nodes are alive, using EDAP can gain an increase of about 75% in network lifetime (as it can be figured from Figure 6(b)). Using LFI method (OHPP) also increases network lifetime about 25% in our simulations by using 6 sources (Figure 6(c)) but this gain can be increased in large-scaled networks up to 50%.

6 Related works

Classical multi-path routing and clustering protocols have been proposed for both wired and wireless networks. The multi-path routing approaches have been used to perform load-balancing and fault tolerance. Alternate path routing schemes in ad-hoc networks have been investigated like TORA (Nasipuri and Das, 2000), which provides multi-path by maintaining a destination-oriented DAG for each node in the network but the overhead for maintaining the DAG in the network is significant. Multi-path extensions to DSR (Park and Corson, 1997) support the construction of alternate paths using the source routing mechanism. This works use disjoint path from intermediate nodes on the primary path to enhance resilience. Taheri Javan et al. (2009) proposed a multi-path protocol based on DSR, which utilises common omni-directional antennas, rather than directional ones and transfers data through multiple zone-disjoint paths simultaneously.

In ad-hoc networks where the energy efficiency is not as restricted as sensor networks and the number of nodes is limited to a few hundred, the multi-path routing overhead is usually negligible. Because of data-centric and localised routing algorithms, used in WSNs, multi-path routing proved to be a complex problem because in such environments, each node can have a local view of the network and no information about the location of the source or the destination is available. Nasiri Eghbali et al. (2009) described the use of Multi-Sink Routing Algorithm (MSDD) for implementing load-balancing and increasing the energy efficiency of the routing algorithm in WSN.

In Ganesan et al. (2001), two methods have been proposed for construction of disjoint and braided paths. This work focuses on increasing the resiliency of sensor networks and load-balancing issues have not been considered. We called their disjoint routing protocol as simple multi-path routing method. Our proactive multi-path routing method in its basic form behave similarly as their work but their definition used for the braided paths are quite different from ours. The braided paths are defined in their work as the set of paths, each excluding a specific node in the primary path, constructed by the DD algorithm. The Gear algorithm (Yu et al., 2001) relies on geographic information to forward the interest packets in the desired location. Energy efficiency is gained in this method during path selection phase and by reducing the flooding traffic of the interest packets all through the network. But, they still use the classic DD algorithm in a more limited area. Our proposed multi-path algorithms can be combined easily with this algorithm to produce multiple paths in the interested area. Also, in Yu et al. (2001), two types of real-time and best effort gradients have been proposed. They used RT gradients to reduce the ETE delay in delay-sensitive applications and BE gradients for performing load-balancing and increasing the energy efficiency of their routing protocol. Although EDDD routing protocol can produce different paths for different kinds of packets, the quality and the quantity of such paths has not been discussed in their work. We also used the idea presented in Braginsky and Estrin (2002) for RFI improvement method.

Heinzelman et al. (2000) proposed LEACH protocol, which is the base idea used in many other clustering schemes. This protocol consists of rounds in which cluster heads are selected probabilistically in the beginning. The cluster head gathers cluster data using TDMA scheduling and send aggregated data directly to sink. Improvements to Heinzelman et al. (2000) are proposed in TEEN (Manjeshwar and Agrawal, 2001) and PEGASIS (Lindesy and Raghavendra, 2002). In TEEN, soft and hard thresholds have been used in each node for sending data to decrease the data transmission rate. PEGASIS is a chain-based power-efficient protocol that forms the chain towards the base-station greedily using geographical information of each node. The chain leader aggregates the data and sends them towards the sink. Cluster heads are not distributed evenly in LEACH. MECH tries to address this problem by gathering local neighbour information to assign cluster heads. Cluster heads in this protocol route data among cluster heads to reach the base station instead of sending them directly to the sink.

7 Conclusion

In this paper, Efficient Data-gathering Protocol (EDAP) for DD algorithm is proposed for situations where an event can trigger more than a single sensor node. We used a VS node near the sources. This node gathers information sent by nearby sources and undertakes responsibility of sending them towards the sink. This way, extra ED packets will be omitted and data from different sources will be aggregated as soon as possible.

Although improvements gained by early-aggregation have not been considered in our simulations, results show that routing overhead will be decreased significantly. This improvement is gained through omitting the extra ED packets flooded by each source. Also, connection lifetime between sources and sink will be increased using proper VS refresh periods.

This paper also describes the use of multi-path routing method for implementing load-balancing and increasing the energy efficiency of the routing algorithm. In this work, proactive methods were proposed for constructing multiple paths. Limited Forward approach is presented to improve the efficiency of this algorithm. The LFI method also decreases the number of ED packets flooded in the network, significantly and reduces the overhead of TPP algorithm. Simulation results also show that using multi-path routing algorithm leads to longer connection lifetimes.

Each of these algorithms has its own weak points. When using ODCP, all paths from different sources will merge after a few hops and all of them are forwarded by VS node towards the sink, using a single path. As mentioned earlier, this may cause the network to be partitioned. Also, proactive multi-path routing algorithm constructs disjoint paths between sources and sink nodes and prevents aggregation. EDAP is introduced to solve these problems by using the benefits of both algorithms and simulation results show that using EDAP can lead to 75% improvement in network lifetime in comparison with DD. However, using multi-path routing technique in the EDAP algorithm may not increase the connection lifetime considerably. The reason behind this is that VS node will die before other nodes and breaks the connection but in real-world applications usually the nodes near the sink node will die before the other nodes and by using EDAP, the lifetime of these nodes will be increased by performing load-balancing techniques.

References

Braginsky, D. and Estrin, D. (2002) 'Rumor routing algorithm for sensor networks', *Proceedings of the 1st ACM international Workshop on Sensor Networks and Applications*, Atlanta, Georgia, pp.22–31.

Ganesan, D., Govindan, R., Shenker, S. and Estrin, D. (2001) 'Highly resilient energy-efficient multipath routing in wireless sensor networks', *Proceedings of the ACM MOBIHOC*, Long Beach, California, USA, pp.251–253.

Heidemann, J., Silva, F. and Estrin, D. (2003) 'Matching data dissemination algorithms to application requirements', *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, California, pp.218–229.

Heidemann, J., Silva, F., Yu, Y., Estrin, D. and Haldar, P. (2002) *Diffusion Filters as a Flexible Architecture for Event Notification in Wireless Sensor Networks*, Technical Report ISI-TR-556, University of Southern California, Information Sciences Institute, USA.

Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2000) 'Energy efficient communication protocol for wireless microsensor networks (LEACH)', *Proceedings of the 33rd Hawaii International Conference Systems Science*, Maui, Hawaii, Vol. 8, p.8020.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000) 'Directed diffusion: a scalable and robust communication paradigm for sensor networks', *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks*, Boston, Massachusetts, pp.56–67.

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J. and Silva, F. (2002) 'Directed diffusion for wireless sensor networking', *ACM/IEEE Transactions on Networking*, Vol. 11, No. 1, pp.2–16.

Lindesy, S. and Raghavendra, C. (2002) 'PEGASIS: power efficient gathering in sensor information system', *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, pp.1–6.

Manjeshwar, A. and Agrawal, D. (2001) 'TEEN: a routing protocol for enhanced efficient in wireless sensor networks', *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, San Francisco, California, pp.2009–2015.

Nasipuri, A. and Das, S.R. (2000) 'On-demand multipath routing for mobile wireless networks', *ACM/Communications*, Vol. 43, pp.51–58.

Nasiri Eghbali, A. and Dehghan, M. (2007) 'Load-balancing using multi-path directed diffusion in wireless sensor networks', *Proceedings of the Third International Conference Mobile Ad-hoc and Sensor Networks*, Beijing, China, pp.44–55.

Nasiri Eghbali, A., Sanjani, H. and Dehghan, M. (2007) 'ODCP: an on-demand clustering protocol for directed diffusion', *Proceeding of the 6th International Conference ADHOC–NOW*, Morelia, Mexico, pp.226–236.

Nasiri Eghbali, A., Taheri Javan, N. Dareshoorzadeh, A. and Dehghan, M. (2009) 'An energy efficient load-balanced multi-sink routing protocol for wireless sensor networks', *Proceedings of the 10th IEEE International Conference on Telecommunications*, Zagreb, Croatia, pp.229–234.

Park, V.D. and Corson, M.S. (1997) 'A highly adaptive distributed routing algorithm for mobile wireless networks', *Proceedings of the INFOCOM 97*, Kobe, Japan, pp.1405–1413.

Taheri Javan, N., Dareshoorzadeh, A., Soltanali, S. and Ghanbari Birgani, Y. (2009) 'IZM-DSR: a new zone-disjoint multi-path routing algorithm for mobile ad hoc networks', *Proceedings of the 3rd European Symposium on Computer Modeling and Simulation*, Athens, Greece, pp.511–516.

Yu, Y., Estrin, D. and Govindan, R. (2001) *Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks*, UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, Los Angeles, California.

Bibliography

Chen, M., Kwon, T. and Choi, Y. (2006) 'Energy-efficient differentiated directed diffusion (EDDD) in wireless sensor networks', *Computer Communications*, Vol. 29, No. 2, pp.231–245.

Chang, R.S. and Kuo, C.J. (2006) 'An energy efficient routing mechanism for wireless sensor networks', *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, Vienna, Austria, Vol. 2, pp.308–312.

Ganesan, D., Cerpa, A., Ye, W., Yu, Y., Zhao, J. and Estrin, D. (2004) 'Networking issues in wireless sensor networks', *Journal of Parallel and Distributed Computing (JPDC), Special issue on Frontiers in Distributed Sensor Networks*, Elsevier Publishers, pp.799–814.

Handziski, V., Köpke, A., Karl, H., Frank, C. and Drytkiewicz, W. (2004) 'Improving the energy efficiency of directed diffusion using passive clustering', *Proceedings of the Wireless Sensor Networks, First European Workshop*, Berlin, Germany, pp.172–187.

Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D. and Ganesan, D. (2001) 'Building efficient wireless sensor networks with low-level naming', *Proceedings of the Symposium on Operating Systems Principles*, Alberta, Canada, pp.146–159.

Heinzelman, W.R., Kulik, J. and Balakrishnan, H. (1999) 'Adaptive protocols for information dissemination in wireless sensor networks', *Proceeding of the ACM MobiCom99*, Seattle, pp.174–185.