Braided Disjoint Branch Routing Protocol For WSNs

Azeddine Attir, Abdelmadjid Bouabdallah, Yacine Challal, Abdelkrim Hadjidj

Braided Disjoint Branch Routing Protocol For WSNs

Azeddine Attir
Université de M'sila
Département STIC
M'sila 28000, Algérie
azeddine.attir@gmail.com

Abdelmadjid Bouabdallah
Univ. Tech. Compiègne
HEUDIASYC UMR 7253
Compiègne, France
bouabdal@utc.fr

Yacine Challal
Univ. Tech. Compiègne
HEUDIASYC UMR 7253
Compiègne, France
ychallal@utc.fr

Abdelkrim Hadjidj
Univ. Tech. Compiègne
HEUDIASYC UMR 7253
Compiègne, France
ahadjidj@utc.fr

***Abstract*—Wireless sensor networks operate typically in a Low power Low connectivity environment. Indeed, communications in such networks are unreliable because of nodes failures (battery depletion, physical damage, etc.), and loss of connectivity (obstacles, interference, etc.). The failure of sensor nodes should not affect the overall operation of the sensor network. In other words, the WSN design should be fault tolerant. At the routing level, disjoint and braided multipath routes are the most used approaches to increase fault tolerance in WSN . In this work, we exploit the advantages of the two approaches and propose a flexible method to construct a maximum number of paths. We demonstrate through simulations, that our solution outperforms the state-of-the-art solutions in terms of the number of paths per node and hence in terms of mean time to failure (MTTF).**

I. Introduction

A wireless sensor network is comprised of a large number of battery-limited sensor nodes communicating with unreliable radio links. Independently of WSN applications, the final objective of the sensor nodes is to sense and send the data to one or more concentrating points called sinks. The short range induced by the radio interface, due to energy limitations, does not allow the nodes to send data directly to the sink. Therefore, all sensor nodes must collaborate to accommodate the objective. One can summarize this collaboration in two points:

- Sensor nodes exchange topology information to form a backbone to reach the sink. This phase is called topology construction.
- Each node must take care to send its data and also retransmit the data generated by other nodes in the network. This phase is called topology exploitation.

In this paper; we will be interested in developing the first item.

The simplest but not fault tolerant approach is to construct only a single path between sensors that sense data and the sink that collects this data. A single link/node fault causes the breakage of communication between one or many nodes and the sink. This is the major flaw of this approach.

To construct a reliable wireless sensor network topology, multipath routing is used. This category of protocols provides tolerance of faults and increases the network resilience. Different multipath schemes have been proposed so far, offering different levels of reliability and fault tolerance. Among these schemes, building node-disjoint paths has been considered as the most reliable one, due to the absence of common sensors between node-disjoint paths, a link disconnection will cause at most a single path to fail for any sensor in the network. This can contribute greatly in prolonging the network lifetime since failures do not cause a significant impact onto the routing view of sensors [1]. Despite the advantages of this approach, the constraint of making all paths disjoint - which is impossible in some situations-, may induce scalability issues.

To overcome this disadvantage, braided multipath routing protocols have been proposed : in such scheme the paths are constructed while tolerating some common nodes or links [2]. However this approach provides lower fault tolerance. Indeed, a failure of a node belonging to several paths will cause the failure of all those paths and may disconnect a large part of the network.

Node-disjoint multipath routing provides a high reliability since a node failure would not cut all paths between a source and the sink. This is the ideal multipath settings. Nonetheless, building multiple node-disjoint paths between sources and the sink requires a high density network. If this configuration could be realistic in the "first age" of a WSN life, this is hardly reachable later when the network experiences node failures and battery depletions which decrease its density. The main idea of our solution is to accommodate network configuration evolution over time through an adaptive multipath construction approach: in the first stage of network deployment our approach builds completely disjoint paths taking advantage of high network density and available resources. This will induce a high workload at sink neighbors. To avoid their rapid battery depletion, our adaptive approach weakens the disjointedness condition and allows paths intersection at downstream nodes with highly available resources (energy in particular). This will shift the workload to more powerful nodes and hence leverages paths quality for longer overall network longevity.

This paper is organized as follows. First, we present an overview of related works. Second, we describe the proposed protocol. Third, via simulation we discuses and evaluate the performances of the proposed protocol. Finally, we conclude the paper.

II. Related Works

View the importance of multipath routing there has been a host of research works in this field in the last few years. Besides improving network resilience, multipath routing is also used for load balancing [3] and QoS provisioning [4]. Using multipath routing provides tolerance of node failures along any individual path and increases the network resilience. Node-disjoint multipath routing protocols construct paths with no common nodes/links. This leads to high resilience and fault tolerance since a node failure will threat only one path. However, they usually suffer from control message overhead and a lack of scalability. In [5], authors proposed a Node-Disjoint Parallel Multipath Routing (DPMR) algorithm. DPMR uses source delay and one-hop response mechanisms to construct multiple paths simultaneously. To ensure node-disjointness, only nodes that have not been occupied by other paths forward route requests to their neighbors. In [6], authors described LAND, a Localized Algorithm for finding Node Disjoint paths. LAND constructs a set of minimum cost node-disjoint paths from every node to the Sink.

Branch aware routing [7] represents an efficient multipath discovery method based on flooding. BRP [7] tags route messages with Sink neighbors IDs (roots) and flood these messages to the network. Upon receiving several requests, a node choses only one branch and forwards it to its respective neighbors. The main drawback of this method is the limited number of discoverable paths. To find more alternative paths, BRP defines a multipath extension flooding phase where nodes from different branches exchange their discovered paths. As a result, BRP discovers more disjoint paths at the cost of more messages exchange. Instead of tagging routes with the roots'ids, in SMRP [1], the tagging responsibility will be assigned to the neighbors of root nodes, which will increase the number of alternative routes.

Some researchers aimed to reduce node-disjoint protocols overhead by relaxing the disjointness requirement; they argue that the construction of partially disjoint paths can reduce the energy consumption and control overhead. In [2], Ganesan et al. explored disjoint and braided paths and compared their performances. They showed that braided path protocols overhead is only half the overhead induced by node disjoint protocols. However, partially disjoint paths are weak since a single node failure causes a broad failure. NC-RMR [8] constructs disjoint and braided multipath to increase the network reliability. Furthermore, it uses network coding mechanism to reduce packet redundancy when using multipath delivery.

In wireless sensor networks, data is forwarded by nodes and routed to the Sink. Thus, nodes nearer the Sink relay more packets and actively participate in communication. As a result, these nodes expand more energy and are more failure prone due to battery depletion. Considering this fact, some works focused the disjointness only where it has the higher impact. SAR (Sequential Assignment Routing) algorithm [9] requires disjointness only in one hop sink neighborhood. To do this, SAR constructs trees departing from each Sink's neighbor by successively branching at each hop. At the end, most nodes will be part of several trees and have multiple paths disjoint inside the Sink one hop neighborhood. To ensure fault tolerance and failure recovery, SAR implements a localized path restoration mechanism by means of messages exchange between sensors. This leads to an overhead and scalability issues.

In what follows, we present the two version of our protocol called Contrary to BRP and SMRP, our protocol reacts to the health of the network and dynamically selects the root nodes that are eligible to be points of intersection; therefore it creates the height number of paths for each node with or without the presence of failures.

III. Version 1:Braided Disjoint Branch Routing Protocol -*BDBRP*-

Researchers have proven that the multi and disjoint paths routing protocols are the best to use in terms of reliability, security and fault tolerant [1]-[2]-[10]. However most of the proposed solutions suffer from scalability issues due to the communication overhead: number and size of control message. Furthermore, as far as we can say, none of the solutions of the literature takes into account the dynamic aspect of the network.

In BRP [7], the exchanged RREQ messages are tagged with the identifier of the first relaying node after the base station. They call these nodes root nodes, and their sub-trees branches. Using these tags, a sensor can easily decide whether two RREQ came from disjoint routes by comparing their branch id.

Instead of tagging routes with the roots'ids, in SMRP [1], the tagging responsibility will be assigned to the neighbors of root nodes, i.e. 2-hops neighbors of the base station. By adding this second level tagging, they allow root nodes as the solely intersection points between routes. Neighboring nodes of roots can become sub-roots and thereby construct their own sub-branches. A sensor will accept paths within the same branch only if they come from different sub-branches, which will increase the number of alternative routes.

We observe that, it's the number of root nodes that governs the number of disjoint paths, i.e., if the number of root nodes is large; the number of disjoint paths is large too. In [7], only the base station neighbors are considered as root nodes, therefore, if the number of these nodes is small, the number of disjoint paths will significantly decrease for all nodes in network.

We know that the majority of traffic passes through neighboring nodes of the base station and especially the closest neighbors (one and two hops), this makes these nodes most vulnerable to the failures and congestion. In [1]-[7] if the number of one and two hops neighbors of the base station is small, the other nodes may not able to discover disjoint paths, and therefore do not tolerate failures of the neighboring nodes of the base station.

Hereafter, we designate by root nodes, nodes that are eligible to be points of intersection. In other words, paths are tagged with root ids and hence are disjoint between sensors up to the roots and braided between the roots to the sink.

In BDBRP, only the one hop neighbors of the base station are being considered as root nodes. Consequently, at the beginning of network deployment, all nodes will have disjoint paths to the base station. In order to avoid rapid battery depletion of root nodes, after a period of time, two hops neighbors of the sink will be considered as root nodes in turn. At the i-th round of protocol execution, the i-th hops neighbors of the sink will be elected as root nodes, and so forth. This dynamic election of root nodes will balance the load of root role over all the nodes of the network and hence increase network longevity. We notice that our solution combines two approaches : disjoint-paths and braided-paths. Indeed the network is configured according to the first approach and converges to the second gradually while accommodating nodes resources. Therefore, our solution leverages, dynamically, path disjointness for residual energy in order to increase the overall network lifetime.

This process is explained in Fig. 2.

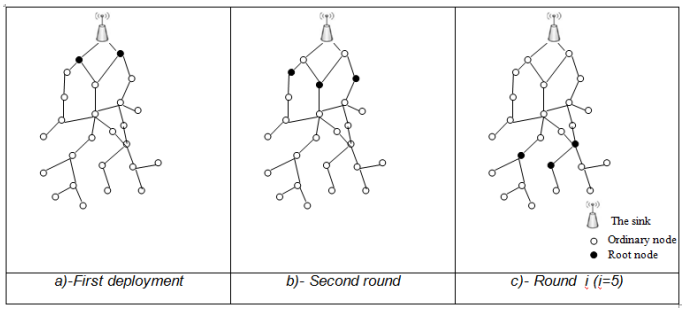

Fig. 1. Root nodes election

A. Protocol description

Each sensor saves a routing table, each entry containing a fresh alternative path that has the following format:

$$\prec ID\_Parent,\ ID\_root\ ,Nbr\_hops \succ$$

indicating respectively the ID of the parent , the ID of its root and the number of hops separating this sensor from the sink. The ID_root field is equal to $\emptyset$ for all sensors between the sink and roots : meaning that all nodes in this region will not be considered as root nodes.

Unlike BRP, in BDBRP it is not required to store the nodes IDs in the path which allows to remove the path attribute. Upon receiving a RREQ, a node compares between the $\prec parent,\ root \succ$ attributes of routing table and the $\prec ID\_Parent,\ ID\_root \succ$ of RREQ. Then, the node can decide if tow paths are disjoint, in which case the new path is inserted into its routing table.

1. Round initialization:

To discover paths relating each sensor node to the sink, at each round the sink broadcast a Route REQuest (RREQ) message having the following format:

$$\langle\ r,\ parent,\ root,\ hopcount \rangle$$

Where :

- r : the sequence number identifying simultaneously the current round and the number of hops separating roots from the sink in the current round.
- *parent* : the ID of the sending node.
- *root* : the ID of the root.
- *hopcount* : the number of hops to the sink.

periodically, the sink starts the construction of a new tree by broadcasting the following message:

$$sink \rightarrow * : \langle r,sink,\emptyset,1 \rangle$$

In the first round, the sequence number is initialized to 1, which means that the one hop sink neighborhood will be considered as roots.

2. Upon receiving RREQ by sensor s:

At each reception of the RREQ message Msg:

2.1. *If (RREQ.r $>$ r in the routing table)* {the sensor initializes its routing table by removing any discovered path}

2.2 *If (RREQ.hopcount $<$ RREQ.r)*

{

// This is the case of sensors between the sink and the roots.

- s takes the best route in term of hop count to the sink and adds it to the routing table with :

- *ID_Parent = RREQ.parent*
- *ID_root* = $\emptyset$
- *Nbr_hops = RREQ.hopcount*

- The node selects an entry with minimal Nbr_hops from its routing table and forward the RREQ with the following values:

$$s \rightarrow * : \langle\ r,\ s,\emptyset,{+}{+}\ hopcount \rangle$$

}

2.3 *If (RREQ.hopcount == RREQ.r)*

{

s becomes a new root, takes the best route in term of hop count to the sink , adds it to the routing table with hop count of routing table equal to *RREQ.hopcount* and forwards the following RREQ:

$$s \rightarrow * : \langle\ r,\ s,s,{+}{+}\ hopcount \rangle$$

}

2.4 *If (RREQ.hopcount $>$ RREQ.r)*

{//This is the case of sensors in the roots downstream.

Upon receiving sub-sequent RREQ messages in the same round, the sensor should verify their intersection with already discovered paths. If the received sub-branch tag does not exist in the routing table, the sending node is selected as an alternative parent and the new route is added to the routing table. Otherwise, the message is ignored since it does not fulfill the required quality.

The node s start a timer.

The node selects an entry with minimal *Nbr_hops* from its routing table and forwards the following RREQ:

$$s \rightarrow * : \langle\ r,\ s,rootid,{+}{+}\ hopcount \rangle$$

where *rootid* represents the ID of the root of the selected entry.

}

The timer in the protocol description is a random decision timer that defines the discovery period of alternative paths before relaying the RREQ message. Timer is only used for sensors under the roots, those sensors must wait to choose the best roots among the received RREQ and relays this decision to its neighborhood. For sensors above the roots we don't need timer hence we accelerate the network formation.

Before proceeding to the simulations and performance evaluations, we present in the next section the second version of our protocol called FDBRP for fully DBRP.

IV. Version 2: Fully Disjoint Branch Routing Protocol -*FDBRP*-

In FDBRP, only the one hop neighbors of the base station are being considered as root nodes. Consequently, at the beginning of network deployment, all nodes will have disjoint paths to the base station. In order to avoid rapid battery depletion of root nodes, after a period of time, two hops neighbors of the sink will be considered as root nodes in turn. At the i-th round of protocol execution, the i-th hops neighbors of the sink will be elected as root nodes, and so forth. This dynamic election of root nodes will balance the load of root role over all the nodes of the network and hence increase network longevity. We notice that to ensure the fully disjointness of paths, in all rounds the one hop neighbors of the base station are considered as root nodes; hence, we distinguish between two sets of root nodes; the first set is called : $roots_1$, this set is formed by the one hop neighbors of the sink and the second set is called $roots_i$ and it is formed by the i-th hops neighbors of the sink at the i-th round, then, nodes between $roots_1$ and $roots_i$ are tagged with $roots_1$ and nodes under $roots_i$ are tagged with $roots_i$.

This process is explained in Fig. 2.

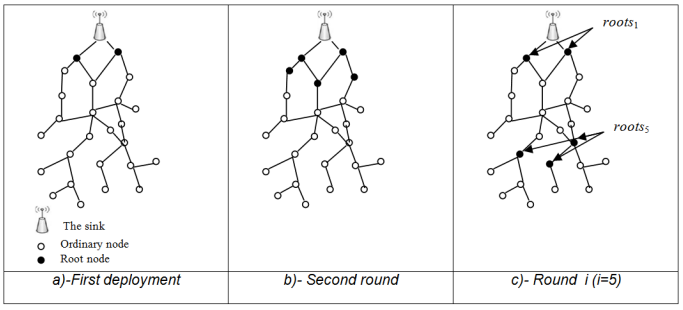

Fig. 2. Root nodes election

A. Protocol description

In this section, we present only the points that differentiate this version from BDBRP; the message format and round initialization is such in BDBRP III-A.

2. Upon receiving RREQ by sensor s:

At each reception of the RREQ message Msg:

1. *If (RREQ.r > r in the routing table)* {the sensor initializes its routing table by removing any discovered path}

2. *If (RREQ.hopcount == RREQ.r || RREQ.root =∅)*

{ This case is verified when the fist condition is checked as true, this means that *s* is a part of $roots_i$, or the second condition, this means that *s* is a part of $roots_1$

s becomes a new root, takes the best route in term of hop count to the sink , adds it to the routing table with hop count of routing table equal to *RREQ.hopcount* and forwards the following RREQ:

$$s \rightarrow * : \langle\ r,\ s,s,++\ hopcount \rangle$$

}

3. *If (RREQ.hopcount<RREQ.r||RREQ.hopcount>RREQ.r)*

{

//Here we distinguish between two cases :

Case 1 : When the first condition is checked, sensor *s* is between the $roots_1$ and the $roots_i$, hence, *s* will be tagged with nodes from $roots_1$.

Case 2 : When the second condition is checked, this is the case of sensors in the $roots_i$ downstream, hence, *s* will be tagged with nodes from $roots_i$.

In any case, upon receiving sub-sequent RREQ messages in the same round, the sensor should verify their intersection with already discovered paths. If the received sub-branch tag does not exist in the routing table, the sending node is selected as an alternative parent and the new route is added to the routing table. Otherwise, the message is ignored since it does not fulfill the required quality.

The node *s* start a timer.

The node selects an entry with minimal *Nbr_hops* from its routing table and forwards the following RREQ:

$$s \rightarrow * : \langle\ r,\ s,rootid,++\ hopcount \rangle$$

where *rootid* represents the ID of the root of the selected entry. }

V. Performance evaluation and simulations

In this section, we simulate and compare the two version of our protocol with BRP and SMRP. Hence, we have implemented BDBRP, FDBRP protocols using networkx , a graph library for python[13]. In addition to our protocol, we have implemented BRP and SMRP protocols which we previously introduced. The network topology generation parameters are as follows: (i), the network area size is 50 X 50 units, (ii) the number of nodes is varied from 200 to 600 nodes (iii) each node has a communication range of 2 units, the Sink node is located in the corner of the network. We note that for the two version of our protocol, we vary the number hop from which the node are considered as root nodes from 1 to 15 and compute the mean of a calculated metric.

A. The number of root nodes

We have explained in section III, that the number of disjoint paths is governed by the number of root nodes. In Fig. 3 we vary the number of nodes in the network from 200 to 600 nodes and for each network size, we have generated a grid network topology, on each topology, we have executed 100

times all protocols and calculated the average number of root nodes (Y axis).

Fig. 3, compared with BRP and SMRP, our two version of protocol present is the highest number of root nodes for every network size. We remark also that the FDBRP version present a greater number of root node then BDBRP, car in FDBRP the root nodes is composed of the sensors at 1 and i hops from the sink whereas, it composed only by i hops from sink for BDBRP. (See protocol description IV-A). Another remarkable point is that the number of root nodes remains constant for most cases; this is due to the static position of the sink, since for every network size it has the same neighbors.

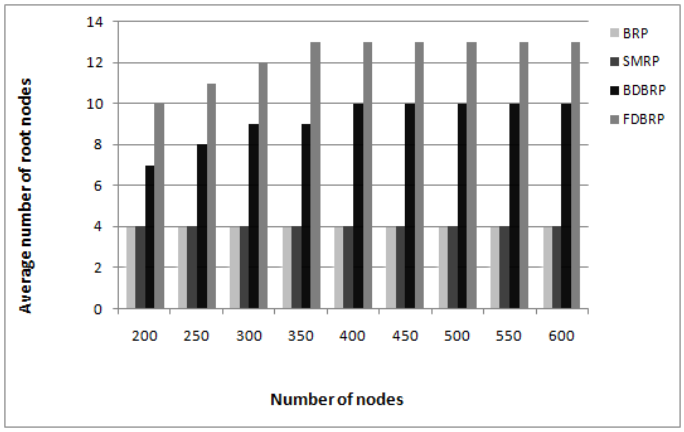

Fig. 3. Average number of root nodes with grid topology

B. The number of paths

The number of constructed paths between a node and the Sink is an important metric to estimate the fault tolerance of our solution. The higher the number of constructed paths, the better the fault tolerance is. For each protocol and each scenario, we measured the average number of discovered paths per node (Y axis) depending on the number of nodes in the network (X axis). For each network size, we have generated a grid network topology and calculated the average number of discovered paths. the Sink node is located in the center of the network. Fig. 4 illustrates the average number of discovered paths between a node and the Sink depending on the network size. We notice that for each network size, FDBRP ans BDBRP discover more paths than BRP and SMRP.

C. Fault Tolerance

To evaluate the fault tolerance of BDBRP and FDBRP, we have considered three metrics : network connectivity, the mean time to failure and the mean time de energy failure.

1) network connectivity: We have considered the impact of node failure on the network connectivity. We have first run BRP, SMRP, BDBRP and FDBRP protocols to construct multiple paths on different network topologies. Then, we have varied the node failure rate from 3% to 15% and computed the number of nodes that still have a functional path to the sink (connected nodes). For each protocol, we have executed

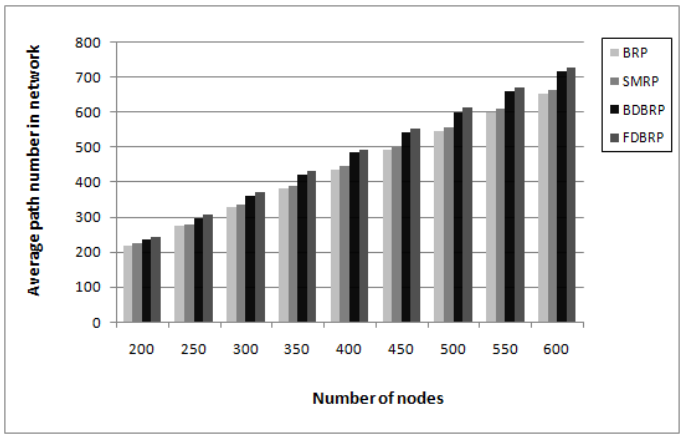

Fig. 4. Average path number in network with grid topology

100 simulations to estimate the average connected nodes rate (Y axis) depending on node failure rate (X axis). The failures are randomly distributed on network nodes and the number of nodes is 600.

Fig.5 illustrates the average number of connected nodes depending on nodes failure rate. We notice that compared to BRP and SMRP, BDBRP and FDBRP are more fault tolerant.

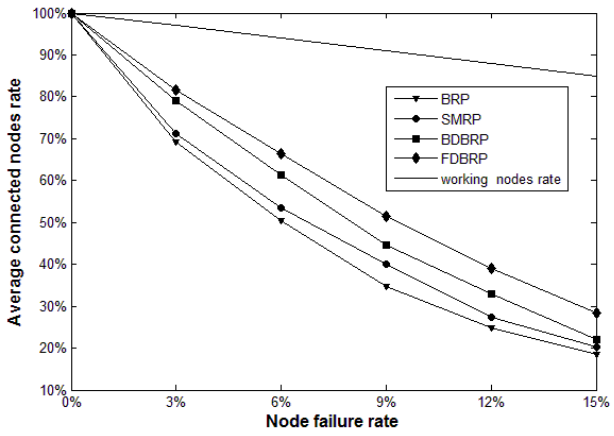

Fig. 5. Average connected nodes rate when network size is 600, BRP, SMRP, BDBRP

2) Mean Time To Failure: The Mean Time To Failure (MTTF) represents an important metric to estimate the contribution of a solution to improve the network lifetime. It is defined as the average period of time during which a system is considered functional and can deliver sensed data to the sink. Applying this definition, we have considered that a routing topology is not functional when some sensors become incapable of reaching the sink. At this time, a reconstruction of the communication topology is necessary to repair the system. Thereby, the MTTF gives also an estimation of the required interval between two tree constructions. This estimation represents precise information to network designers for establishing

an optimal schedule of topology creation.

To evaluate this metric, we have simulated the protocols to obtain the constructed routing topologies while considering a grid network topology. With the resulting routing topologies, we have simulated failures of nodes as a Poisson process with a rate of 2 failures per unit of time. When a failure occurs, we randomly select an active sensor from the network and remove it from the topology. Afterward, we verify whether the resulting graph is still connected to simulate a new failure. In the case of a disconnected graph, the system is considered "not functional" and the summation of the intervals between failures gives the time to failure. To estimate the MTTF, we considered the average of 1000 iterations for each simulation scenario and calculated the 0.96 confidence interval for each point. The confidence interval is plotted as a bar error surrounding the average value.

Fig.6 and Fig.7 present the simulation results.

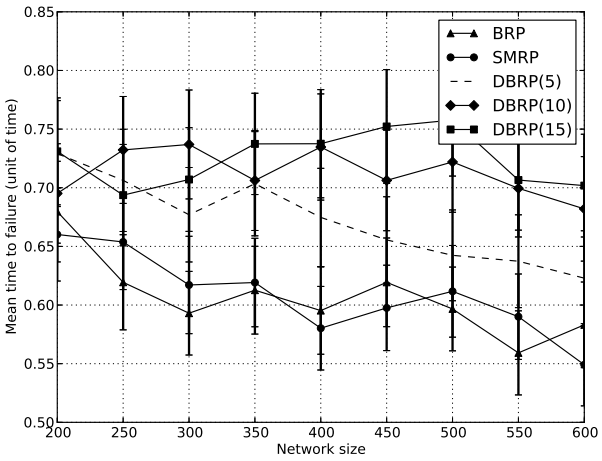

Fig. 6. Mean Time To Failure for BDBRP

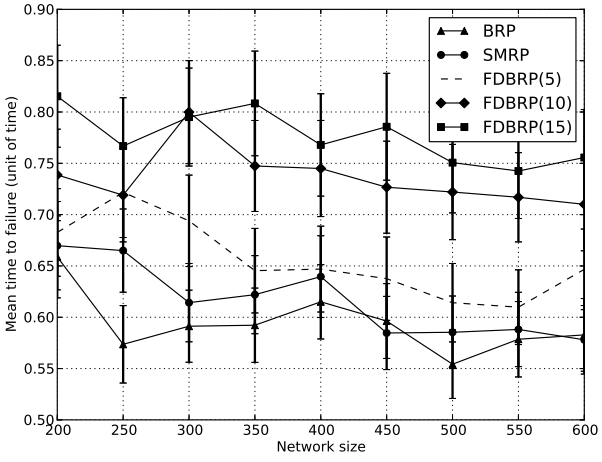

Fig. 7. Mean Time to Failure for FDBRP

For both versions of our protocol, we set as roots, nodes that are at 5,10 and 15 hops from the base station and we have computed the MTTF. Fig.6 and Fig.7 show the results. The first point to be noted is the small value of MTTF, this is mainly due to the grid topology in which the degree of nodes is very low. With such a degree, the number of disjoint paths is very low which leads to a small MTTF. Nevertheless in such non fault tolerant topology, the two versions of our protocols have the batter performance. We remark also, that if we set root distance to the sink to 10 and 15 hops we obtain a best performance. This is because, the number of disjoint paths increase with the depth in topology , therefore the failure of a node does not affect all paths and this leads to improve network lifetime in term of MTTF.

3) Mean Time to Energy Failure: For this metric we don't consider a random node failure, but a failure caused by energy depletion. To evaluate this metric, we have simulated the protocols to obtain the constructed routing topologies while considering a grid network. With the resulting routing topology we consider two scenarios for the traffic generation:

Scenario 1: In this scenario all nodes periodically sense and send the data to the sink. For the energy, we take the model discussed in [11]; where energy consumption is mainly due to the transmission and reception of messages. The sink has enough energy so that it never falls down because of its exhaustion. We assume that we have a homogeneous sensor network where each node is equipped with a battery of 1000 units, the quantity of energy consumed during the transmission and reception are set to the values of CC1000 radio configuration [12], when the number of units becomes zero, the sensor node is removed from the network. Afterward, we verify whether the resulting graph is still connected . In the case of a disconnected graph, the system is considered "not functional" and the summation of the intervals between failures gives the time to failure. To estimate the MTEF, we considered the average of 1000 iterations for each simulation scenario and calculated the 0.96 confidence interval for each point. The confidence interval is plotted as a bar error surrounding the average value.

In Fig.8, the MTEF for BDBRP (10) and BDBRP (15) is less then BRP and SMRP protocols , this is explained by the fact that the braided area is large -with all nodes sensing and sending data-, the nodes in such area consume their energy quickly, hence the node in the disjoint area can't find paths to the sink. This is not the case for FDBRP (5). In this configuration, compared to the disjoint area, the braided area is too small, hence we obtain a better MTEF.

For the FDBRP, Fig. 9 shows that for the three root node configurations, FDBRP depicts very good performance. This can be explained by the fact that all paths are disjoint, due to the absence of common sensors between node-disjoint paths. A failure of nodes will cause at most a single path to fail for any sensor in the network; hence, this can contribute greatly in prolonging the network lifetime,therefore, our solution leverages, dynamically, path disjointness for residual energy in order to increase the overall network lifetime.

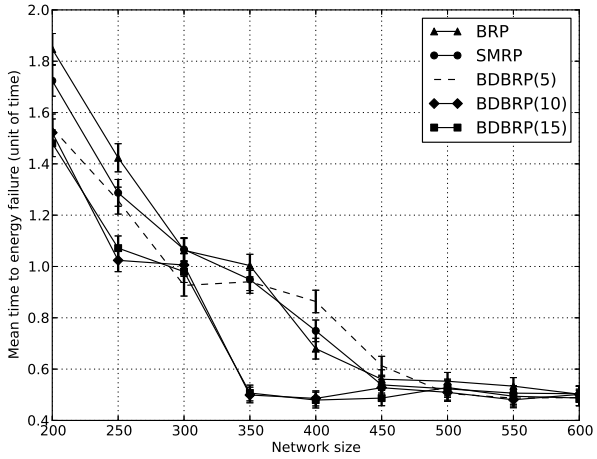

Fig. 8. Mean Time to Energy Failure for BDBRP in continuous sensing network

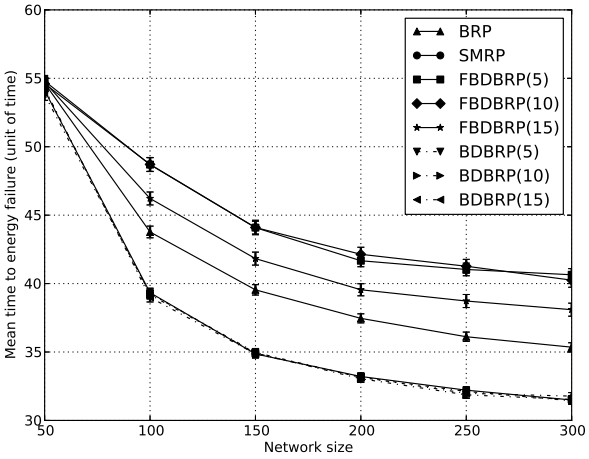

Fig. 10. Mean Time to Energy Failure for event-driven

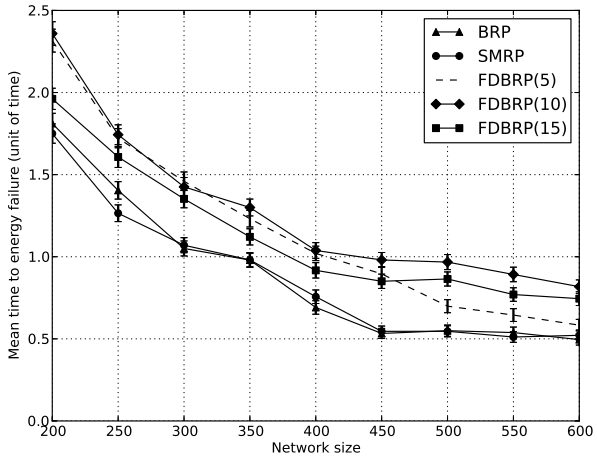

Fig. 9. Mean Time to Energy Failure for FDBRP in continuous sensing network

Scenario 2: Here, we consider an event-driven wireless sensor network application, in which the node that detects an event and needs to send data to the sink is randomly chosen. In this case we suppose that each node is only equipped with 100 unit of energy, we vary the network size from 50 to 300 nodes. To estimate the MTEF, we considered the average of 1000 iterations.

For this scenario, Fig.10 illustrates that BDBRP improves network lifetime and it is competitive with SMRP, but BRP works better. In the other hand, FDBRP shows for the three root node configurations a highest performance.

VI. Conclusion

Multipath routing is a technique that improves fault tolerance. Node disjointness is the main metric used to measurec the quality of discovered paths. Unfortunately, while node disjointness guarantees the best fault tolerance, this requirement reduces the number of possible alternative paths per node, in common topologies. Therefore, many solutions relax this requirement to allow discovering more alternative paths.
In this paper, we proposed a new dynamic approach called Dynamic Branch Routing Protocol that leverages, dynamically, node disjointness for higher number of alternative path per node. We demonstrated through simulations that our protocols outperforms comparable solutions from the literature in terms of fault tolerance without inducing extra energy or message overheads. As a future work, we try to propose a simulation and/or analytical model for the root nodes choice, to have the best performance for deferent network topologies.

References

[1] Y. Challal, A. Ouadjaout, N. Lasla, M. Bagaa and A. Hadjidj. *Secure and Efficient Disjoint Multipath Construction for Fault Tolerant Routing in Wireless Sensor Networks* . Journal of Network and Computer Applications (JNCA), Elsevier, vol. 34, No. 04, pp. 1380-1397, 2011

[2] Ganesan D, Govindan R, Shenker S, Estrin D. *Highly-resilient, energy-efficient multipath routing in wireless sensor networks*. SIGMOBILE Mobile Computing and Communication Review 2001;5:11-25

[3] Kim, E. Jeong, Y.-C. Bang, S. Hwang, B. Kim, *Multipath energyaware routing protocol in wireless sensor networks*, in: 5th International Conference on Networked Sensing Systems, 2008 pp. 127130.

[4] S. Li, R. Neelisetti, C. Liu, A. Lim, *Efficient multi-path protocol for wireless sensor networks*, International Journal of Wireless and Mobile Networks 2(1) (2010) 110130.

[5] S. Li, Z. Wu, *Node-disjoint parallel multi-path routing in wireless sensor networks*, in: Proceedings of the Second International Conference on Embedded Software and Systems (IEEE)2006, pp. 432437.

[6] R. Hou, H. Shi, *A localized algorithm for finding disjoint paths in wireless sensor networks*, IEEE Communications Letters 10 (2006) 807 809.

[7] Lou W, Kwon Y. *H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks*. IEEE Transactions on Vehicular Tech- nology 2006;55:1320-30.

[8] Y. Yang, C. Zhong, Y. Sun, J. Yang, *Network coding based reliable disjoint and braided multipath routing for sensor networks*, Journal of Network and Computer Applications 33 (2010) 422432.

[9] K. Sohrabi, J. Gao, V. Ailawadhi, et G.J. Pottie, *Protocols for self organization of a wireless sensor network*, IEEE Personal Communications 7 (2000) 1627.

[10] A. Hadjidj, A. Bouabdallah, and Y. Challal. *HDMRP : An Efficient Fault-Tolerant Multipath Routing Protocol for Heterogeneous Wireless Sensor Networks*. In Proceedings of the 7th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (Qshine 2010). Houston, USA. November 2010.

[11] Heinzelman, W. R. and Sinha, A. and Wang, A. and Chandrakasan, A. P,*Energy-scalable algorithms and protocols for wireless microsensor networks*. In Proceedings of IEEE International Conference on the Acoustics, Speech, and Signal Processing, ICASSP '00. Washington, DC, USA

[12] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, *A dualradio framework for MAC protocol implementation in wireless sensor networks*, in Proc. 2011 IEEE ICC, pp. 16.

[13] http://www.python.org/