An Event-Detection Estimation Model for Hybrid Adaptive Routing in WSNs

Carlos M.S. Figueiredo†, Eduardo F. Nakamura†, Antonio A.F. Loureiro*, Linnyer B. Ruiz*

*Department of Computer Science, Federal University of Minas Gerais
Belo Horizonte, MG, Brazil
Email: {mauricio,nakamura,loureiro,linnyer}@dcc.ufmg.br
†FUCAPI – Analysis, Research and Technological Innovation Center
Manaus, AM, Brazil

Abstract—A fundamental goal of a wireless sensor network (WSN) is to collect and deliver data to external applications. Due to the strong constraints of these networks, routing algorithms usually consider application-specific characteristics and, consequently, there is no self-contained algorithm appropriate for every case. In particular, many WSN applications are event-driven. In such scenarios, the behavior of the network may vary a lot, which favors different algorithms at different instants. In these cases, hybrid adaptive solutions are more suitable by allowing the adoption of a better routing strategy in response to the variation of the network conditions. In this work, we evolve this approach by considering the temporal characteristic of event occurrence and detection through an estimation model. The proposed solution, called Multi, predicts the need of routing infrastructure creation and maintenance for the adaptation between a reactive and a proactive strategy. To show the advantages of the proposed approach, an instance of Multi is evaluated through simulations. Comparisons with its independent reactive and proactive components and other proposed solutions show improvements on energy consumption.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) [1], [2] have become popular in the research community due to their wide applicability in several areas such as environmental, medical, industrial, or military monitoring [3]. Composed of a large number of low-cost sensor nodes capable to perform data processing and wireless communication, these networks form a powerful platform to monitor areas of interest.

The design of these networks is also challenging. Usually, solutions for WSNs present strong efficiency requirements, because of strong energy restrictions, and limited computational and communication capacity. Also, they demand self-organizing features, i.e., the ability of autonomously adapt to changes resulted from external interventions, such as topological changes (due to failures, mobility, or node inclusion) or reaction to a detected event, without the influence of a centralized entity.

An important issue in WSNs is to collect and process data from the environment and send it to be further processed and evaluated by an external entity connected to a sink node (or gateway). Consequently, routing towards the sink node is a fundamental function. Due to the efficiency requirements, different algorithms have been proposed considering specific application and scenario characteristics [4]. Thus, given a specific scenario, the WSN can be designed to operate with the most appropriated routing algorithm, which can be defined a priori.

Regarding the application characteristics, WSNs can demand different data collecting strategies, such as periodic, eventual, or on-demand [5]. In particular, we expect that several applications follow the event-driven model. These applications are concerned about special events that are not so usual (e.g. fire or intrusion detection), so the network must work in a low activity mode until the occurrence of events for resource savings. In such scenarios, the network behavior may vary a lot in an unpredictable way. For example, the network may remain for a long time with a low or inexistent traffic, but at a given moment a lot of events start to occur increasing the traffic and so the routing needs. This characteristic favors different algorithms at different instants, but it might be unfeasible or undesirable to an external entity to act dynamically on the network to change its behavior, so the network must work in this manner based on autonomic principles.

Hybrid adaptive approaches for routing are interesting solutions to deal with such scenarios by allowing the adoption of the better routing strategy in response to the variation of the network conditions. Some protocols have been proposed for ad hoc networks (MANETs), such as ZRP [6] and SHARP [7]. However, they are not suitable for WSNs due to fundamental differences in the network, applications, and traffic characteristics. A first effort in the WSN domain was made by Figueiredo et al [8], who show the composition of traditional reactive and proactive routing strategies for WSNs in a hybrid solution. However, that solution does not consider the event occurrence characteristic properly.

In this work, we evolve this approach by considering the temporal characteristic of event occurrence and detection through an estimation model. Thus, the proposed solution, called Multi, captures the seasonal and dynamic characteristics of events and predicts the need of routing infrastructure creation and maintenance for the adaptation between the implemented reactive and proactive strategies. The advantages of the proposed solution are shown through simulations, in which comparisons with its independent reactive and proactive algorithms and with other well-known proposals in literature...
show improvements on energy consumption.

The remaining of this paper is organized as follows. Section II extends the discussion on routing in WSNs. Section III presents Multi, our hybrid adaptive solution. Section IV presents the evaluation of Multi through simulations. Finally, Section V presents our final considerations and discusses some future directions.

II. FUNDAMENTALS AND RELATED WORK

Routing in WSNs differs from traditional networks in many aspects. Essentially, energy efficiency is the main aspect considered in these networks due to their very limited and constrained resources. Additionally, a basic goal of a WSN is to collect and process data from the environment and send it to be further processed and evaluated by an external entity connected to a sink node. Consequently, routing towards the sink node is a fundamental task and different algorithms have been proposed [4], each one of them being more suitable for a different case or scenario due to their different features.

Basically, there are the following classes of protocols for WSNs:

Flooding and Gossiping [9]. These are classical mechanisms to forward data in sensor networks that do not need to maintain any routing infrastructure or topology. In the flooding algorithm, every data is sent by broadcasting it to all its neighbors until it reaches the destination. Gossiping differs from flooding by choosing random nodes to forward the data. Although these approaches have no creation and maintenance, they cause data packet implosion, which represents an excessive cost for WSNs.

Proactive protocols. In this class, the routing infrastructure is created and constantly maintained no matter the network behavior. In general, this process is triggered by the destination nodes. Examples of this approach are DSDV [10] for MANETs and various tree-based protocols for WSNs (e.g., One-Phase-Pull Diffusion [11] and some implementations by Woo et al. [12]). This approach can result in improved routing, but it has the disadvantage of a constant resource consumption.

Reactive protocols. In this class, the routing infrastructure is built only when a node wants to transmit a packet. AODV [13] is a well-known protocol for MANETs and Push Diffusion [11] and INFRA [14] are examples for WSNs with such behavior. This approach saves resources in inactivity periods, but has the overhead of path discovery for each originator node.

Another class of routing algorithms consists of the design of hybrid adaptive algorithms, which apply both reactive and proactive strategies that are chosen according to network conditions. Some protocols can be found in the literature for MANETs which compare both independent reactive and proactive approaches. For example, ZRP (Zone Routing Protocol) [6], the first protocol to apply reactive and proactive strategies in a hybrid solution, establishes a zone around every node where routing updates are performed proactively, and outside of these zones the protocol responds reactively. Its main goal is to reduce the routing overhead. SHARP [7] presents an extension of this approach in which zones can be dynamically determined only around the nodes with significant incoming data, and it also allows adaptation with other application-specific metrics, such as jitter and loss rate, in addition to routing overhead.

Such a hybrid adaptive approach has not been fully applied to the WSN domain and the referred MANET protocols are not suitable for sensor networks due to several particularities that must be considered for a proper solution in these networks. Unlike MANETs, in which communication is essentially many-to-many, in WSNs it is generally many-to-one (sources to sink or sources to cluster-head). This characteristic provides a broader view for a sink or cluster-head to perform an adaptive control of the nodes under its responsibility. The traffic characteristic also differs between these networks, and this fact must be considered in an adaptive model. While in MANETs the nodes can dictate how data is generated according to their independent applications and user needs, in WSNs this traffic can be dictated by an external application for the whole network according to different requisition models or queries [5]. Also, for event-driven applications, there may be a spatial and temporal correlation among traffic of different nodes. Again, all these characteristics can lead to a proper adaptive rule in the WSN domain, which is the goal of the Multi solution.

III. MULTI: A HYBRID ADAPTIVE ALGORITHM FOR ROUTING IN WSNs

A. An Overview

Multi is built using traditional reactive and proactive routing strategies for WSNs, and it adapts its behavior autonomously between both strategies in response to the variation of network conditions, in case, event occurrence and detection. Particularly with event-driven scenario, the network can stay inactive or with a low activity for long periods (e.g., months or years) until something happens. To extend the network lifetime, it is mandatory to reduce the energy consumption during the inactive periods. In such cases, the reactive routing approach is preferable because it avoids the proactive maintenance of the routing infrastructure and it is created only when necessary. However, in a given moment, several events may be detected generating a high traffic, which can be appropriated for proactive algorithms to avoid the high path discovery cost of a reactive strategy.

In order to better detect these varying conditions, and properly decide which strategy is more suitable for lower energy consumption, Multi applies an event-detection estimation model and an adaptation rule. Thus, Multi changes the routing strategy from a preferable reactive scheme to a proactive one only if the traffic is expected to increase with new nodes detecting events.

The adaptive control of Multi is performed by the sink node that interacts with external applications and monitors the traffic characteristic of the network, so many-to-one communication is assumed. Clustering, a common approach in WSNs [15] for achieving more scalability and resource savings as the
number of nodes and the sensing area increases, can also be considered. Thus, the cluster-heads can be responsible for coordinating the activities of all nodes in its area, and in our case to perform the adaptive control. The clustering approach will be better evaluated as future work.

The proactive and reactive components of Multi are simple versions of classical approaches for WSNs and it is neither our intention to consider them as new contributions, nor to compare them with other solutions in the literature. Our goal is to show how a hybrid solution can be built from them. Next, we present these components, their integration in the Multi solution, and the applied adaptive model for event-driven scenarios.

B. Proactive Component: EF-Tree

A simple and efficient structure for data collecting in WSNs is a routing tree. It has been evaluated in some studies such as One-Phase-Pull Diffusion [11] and some implementations by Woo et al. [12]. Generally, the tree structure is created and maintained by a sink node in a proactive fashion, and periodical updates (rebuilding) are used to handle eventual topological changes, link problems due to interference or traffic variations, node energy degradation, etc.

Our implementation, called EF-Tree (Earliest-First Tree), works as follows: The sink node starts the process by broadcasting a control message with an unique id (e.g. a sequential number). When a node initially receives the building message with a new id, it identifies the sender as its parent and broadcasts the building message to all its neighbors. Messages received from other neighbors with the same id are discarded. Whenever a node has a data to be transmitted (sensed or forwarded by another node), it will send it directly to its parent. This building process is periodically repeated to update routes, and this periodicity depends on how frequent topological changes occur (e.g., more dynamic networks need shorter rebuilding periods).

Note that it is possible to define other possibilities to choose a parent node by allowing a node to receive several construction messages before choosing its parent. Thus, a parent can be chosen as the node that belongs to a path with the highest amount of available energy, the node closer to the sink, or the node with better link quality (as in [12]), for example. We intend to evaluate some of them as our future work.

C. Reactive Component: SID

Another common approach for routing in ad hoc and sensor networks is to use on-demand algorithms, such as the AODV [13] and Push Diffusion [11]. In this approach, the network may remain inactive until the communication process is started by sensors that have data to be sent, which is appropriate for event-driven scenarios. Obviously, the advantage of this approach is reinforced in situations where no management or control traffic is required, and periodical wake-up schemes, such as STEM [16] or B-MAC [17], can be used to save more resources while no event is detected.

We propose a simple implementation of an algorithm with such reactive behavior called SID (Source-Initiated Dissemination). It is very similar to the Push Diffusion and differs from it by the fact that it performs route discovery only when an event is detected and a route is not available, not periodically as the Push Diffusion. This feature makes SID more reactive in the presence of network dynamics and more energy efficient when the network is inactive. Our implemented SID protocol is presented as follows.

When a source node detects an event and have no valid routes for data forwarding, it starts a data flooding for path discovery with its identification (we use the source node’s unique id). Whenever a node receives this data from its neighbors, it stores the source identification related to the last sender identification (last hop), and proceed with the flooding. Due to this flooding process, nodes will receive the discovery packet from all their neighbors, however, they register and forward only once (the first data received). When the discovery message arrives at the sink node, it will respond with a requisition message to the neighbor node that sent the first data message. This response is recursively propagated by the intermediary nodes towards the data source by using the stored table of sources and last hops. Such requisition messages establish a route from the source towards the sink, and this route inhibits new floodings by considering a validity interval (controlled by timestamps).

In order to allow the network to adjust to eventual topological changes, the requisition messages are periodically sent by the sink towards the sources while data is being received. Once a node (source or intermediate) stops receiving requisition messages, due to any topological change, it will restart to send or forward data in broadcasts. Thus, if any path exists, data will reach the sink again and it will restart the requisition process. Whenever the events disappear, data will not be generated anymore. Consequently, the sink node will stop sending requisition messages to the sources, and the network will become inactive again.

D. Integrating the Strategies

For the hybrid operation, both reactive and proactive strategies must be integrated for the proper network operation. Actually, both strategies share the same goal which is the establishment of a parent for data forwarding towards the sink. Also, in both cases, the routes are determined from control messages propagated by the sink. Thus, as we argue that the reactive strategy is preferable in event-driven scenarios, we integrate both strategies by adopting the reactive one as the basic implementation, and including the possibility of tree construction (following the EF-Tree behavior) instead of sending individual requisition messages according to the sink decision. The resulting behavior is: if there is a parent defined by the SID requisitions or by EF-Tree construction messages inner a validity interval (maintained by a timestamp and their predefined route update period), the data will be forwarded directly to it; otherwise, the data is sent by flooding for path discovery as in SID. The main difference of this integrated so-
olution is that now the sink can respond establishing individual routes as in SID, or proactively create routes for the whole network as in EF-Tree.

E. Estimation Model and Adaptive Rule

In this section, we describe how adaptation is performed with the routing mechanisms described above in order to achieve performance improvements. In this work, our goal is to achieve energy savings.

In event-driven scenarios, a sensor node detects an event when its measured value represents a situation of interest. Once this occurs, we assume that the sensor node starts to generate data to the sink in order to inform and allow monitoring by some application.

In such scenarios, we also expect spatial and temporal correlation in event detections by sensor nodes. Additional to the influence area of events, that causes simultaneous detections in a proximity area, events may present a seasonal characteristic, where they can be distributed in time following an occurrence ratio, or a dynamic characteristic, where they can present an increasing actuation range or mobility (as the examples in Fig. 1). Thus, we can expect that an event with certain characteristic will not change abruptly. For example, a mobile or increasing event will be detected by new nodes following its velocity or increasing rate, or a seasonal occurrence characteristic will be maintained, and so its detection rate.

As described before, reactive algorithms are preferable in inactivity and low occurrence conditions. But by analyzing the reactive and proactive algorithms presented before we can observe that the cost of a data flooding for path discovery in SID is similar to the cost of a tree-building in EF-Tree, which establishes a routing infrastructure for all reachable nodes at once. Thus, if we expect to have at least one source detecting events in the next time interval of route validity, the proactive behavior can be taken to avoid new discovery floodings and, thus, achieve energy savings. This is used by the adaptive model of Multi.

For event-detection estimation, we apply the simple signal processing method of Moving Average Filter (MAF) [18] on the number of new node detections (nodes starting to send data of an event). As the name suggests, this filter computes the arithmetic mean of a number of input measures to produce each point of the output signal, and this can be translated in the following equation:

$$MAF_{out}[i] = \frac{1}{m} \sum_{k=0}^{m-1} input[i+j], \quad (1)$$

where $m$ is the filter’s window (number of input observations to be considered).

We have chosen the MAF method due to its capacity to capture the behavior of the last $m$ monitoring intervals, which is good for seasonal occurrence detection, and due to its characteristic of step response, which is good to detect correlated detection increases. It also filters the noise of measurements (e.g., packet losses, queue delays, or an occasional event distribution), improving the estimation. Other data fusion methods can be used for the node detection estimative, but MAF is simple and has low computational cost, necessary for constrained WSNs.

For the Multi’s adaptation, as soon as a high traffic condition is detected and a proactive behavior is taken, more advantages can be achieved. Thus, we configured the monitoring periodicity (and MAF supply) as equal to the data generation rate of the source nodes (10 s in our simulations). The output of MAF is used as the estimate for the next period of observation. Therefore, as we want the estimate for the next route validity interval (ten times the data rate in our simulation cases), we can assume that it is $10 \times MAF_{out}$. Thus, the proactive behavior will be taken if this estimative is higher than or equal to 1 (i.e., a new estimated detection), and it corresponds to the $MAF_{out} \geq 0.1$ condition.

If Multi is already operating in the proactive mode with a valid routing infrastructure, this action is unnecessary and it is avoided until the validity expires. The resulting adaptive rule of Multi is depicted in Fig. 2.
IV. SIMULATION AND EVALUATION

In this section, the algorithms described in this work are evaluated through simulation. We first present an evaluation of the MAF estimator to setup its parameters. Next, we describe some specific scenarios assessing the performance of a Multi implementation described in this work. We compare Multi with SID and EF-Tree alone, with other well-know algorithms of the literature, which are Push Diffusion and One-Phase-Pull Diffusion, and with the version presented in [8].

A. Parameters

The experiments were performed using the ns-2 Network Simulator\(^1\). The simulation parameters were based on the Mica2 Sensor Node\(^2\): transmission power of 45.0 mW, reception power of 24.0 mW, bandwidth of 19200 bits/s, and a communication radius of 40 m. As this platform uses a CSMA/CA like MAC layer protocol, we used the IEEE 802.11 implementation available in ns-2. In fact, we cannot ignore the power consumption relative to the channel listening (corresponding to idle power), which is very near to the reception power in the Mica2 nodes. But this consumption is very reduced in solutions like B-MAC [17]. Also, as it is equal for all the evaluated algorithms, which results in a constant amount of energy added to all algorithms (verified through simulations), we do not considered it for the comparative analysis.

In all simulations, we considered a network size of 50 nodes randomly distributed in an area of 100 × 100 m\(^2\) and only one sink. Both data and control messages have 20 bytes and are transmitted every 10 s and 100 s, respectively. The main metric evaluated was the energy consumption, which is a restrictive resource in a WSN. All experiments were executed 33 times with a confidence interval of 95% (vertical bars in graphics).

B. MAF estimator evaluation

Due to the absence of a generic event model for WSNs, in this work we represent the event-driven scenario by two types of detection distribution along time: uniform and normal distributions. In the first case, the variation on the detection ratio is represented by different number of sources generating data randomly along the simulation time. This distribution can represent the occurrence of uncorrelated event detections. In the second case, detections are distributed with a standard deviation around an average simulation time, and this distribution can represent the occurrence of correlated detections.

To better understand how the MAF estimator works, and how the window size \(m\) affects its performance we show in Figs. 3 and 4 some simulation cases following the previously described uniform and normal distribution considering 50 source nodes and window sizes of 10 and 50. The graphics show the detection measurements every 10 s, the resulting calculated MAF, and the consequent operating mode of Multi (reactive or proactive mode).

In all cases we can note that the lower \(m\), the faster the variation detection. On the other hand, the greater \(m\), the smoother the estimation. For \(m = 10\), we can see in Fig. 3(b) that MAF does not capture properly the seasonality of detection occurrence in a uniform distribution, causing several swaps between the reactive and proactive modes, and some of them undesirable. This can be seen in the change to reactive mode in the instants around 1100 s, due to an occasional empty interval just before it, which do not reflect the overall behavior of detections. In the sequence, a new detection happens, which represents an additional cost with the reactive behavior, and the Multi turns to proactive mode again. With \(m = 50\) (Fig. 3(c)) these transitions are not so common and MAF has a better performance, however, it takes more time to respond from the initial condition (initial occurrences around 500 s). The impact of \(m\) is also observed with normal distribution in Figs. 4(b) and 4(c), but in this case, a lower \(m\) is better to respond faster to increases in the correlated detections.

Based on more simulation cases with the adopted parameters, we obtained \(m = 20\) as the best value to capture seasonal characteristics of uniform detections, which does not respond so slowly to variations of the normal distribution. This value represents two infrastructure updating intervals (200 s) and it was used in the simulations of Multi presented below.

C. Multi Evaluation

In the first scenario, we have sources generating data randomly with a normal distribution along the time representing the occurrence of correlated detections of events. We set the average of this distribution in the middle of the simulation time of 4000 s and standard deviation of 100 s. The duration of the data generation was a random value between 1 s and 100 s.

Figure 5(a) shows the summary of the consumed energy in the entire network after the specified simulation time considering that the number of source nodes which detected an event varied from 5 to 50 nodes, randomly chosen. In this simulation, with a not very intense traffic, all algorithms delivered nearly 100% of the packets (this graphic was omitted). Regarding the energy consumed, Multi outperforms the other algorithms almost to all number of sources. With the lowest number of sources (5 sources) Multi and SID consume almost the same energy, however, the difference between them becomes near 100% with 50 sources. The advantage of Multi is the adoption of the reactive behavior when the network is inactive, corresponding to energy savings related to EF-Tree, and the flooding avoidance of new sources when the proactive behavior is assumed, which represents a high cost to SID as the number of sources increases. This proactive behavior is taken when the MAF detects an increase on the traffic, so its value becomes higher than the threshold 0.1.

In the second scenario, based on the previous one, we represent uncorrelated detections of events. The variation on detection ratio is represented by different numbers of sources generating data randomly along the simulation time of 4000 s with uniform distribution.

---

\(^1\)http://www.isi.edu/nsnam/ns/

\(^2\)http://www.xbow.com/
and consequently no proactive behavior is taken.

\[ MAF = \frac{m}{10} \]

Routing mode with

Multi always starts to operate as SID, changing to EF-Tree only when the MAF detects this traffic condition, so some initial data flooding happens. In a real event-driven scenario, we can expect to have longer inactivity periods than in the previous simulated scenario. Thus, the energy cost of the EF-Tree is shifted to higher levels due to its proactive behavior.

In this last case, with high number of detections, Multi and EF-Tree performances are not the same because as the simulation time was kept constant, the increase in the number of sources leads to a decrease in the inactivity time amortizing the cost of the EF-Tree from the start of simulation, while Multi always starts to operate as SID, changing to EF-Tree only when the MAF detects this traffic condition, so some initial data flooding happens. In a real event-driven scenario, we can expect to have longer inactivity periods than in the previous simulated scenario. Thus, the energy cost of the EF-Tree is shifted to higher levels due to its proactive characteristic which results in a better relative performance for both SID and, in special, Multi.

To evaluate the resilience of Multi, we fixed the number of sources in 50 nodes with a uniform distribution in the simulation time, and varied the probability of node failures from 0 to 50% randomly during the simulation. When a node fails, it stays inactive until the end of simulation. As we can see in Fig. 6(a), all algorithms presented a similar performance regarding the delivery ratio (relation between received and data packet sent) because in all cases the time to recover from a failure is equal (10 times the data rate). Regarding the energy consumption, instead, the EF-Tree outperforms SID. This happens because in the SID algorithm sources have to flood messages for path discovery whereas the EF-Tree algorithm rebuilds the tree for the entire network at once. Multi’s average performance was between EF-Tree and SID performances because it can operate in one of these two modes during the network lifetime.

In Fig. 5(b), SID and Multi outperform EF-Tree when the number of sources is small (less than 25). In this case, Multi operates just like SID, due to a low number of detections by Multi’s MAF and consequently no proactive behavior is taken. This advantage happens because, with this reactive behavior, the routing infrastructure is created and maintained only when necessary and constant proactive behavior is unnecessary with a low amount of detections. However, when the traffic increases, EF-Tree becomes more adequate because it builds the routing infra-structure for all nodes at once and this proactive behavior is compensated by the initial floodings of SID for path discovery. In this case, Multi’s performance becomes closer to the EF-Tree, outperforming SID. It happens because MAF detects the high traffic condition and the proactive behavior is taken. Again, all algorithms present a high packet delivery ratio and this graphic was omitted.

In Fig. 5(a), SID and Multi present a similar performance which results in a better relative performance for both SID and, in special, Multi.
D. Comparison with other algorithms

To evaluate Multi performance against others contributions found in literature, we have chosen the Push and One-Phase-Pull(1PP) versions of the Direct Diffusion [11]. We used the same scenarios described before and set Push and 1PP Diffusion parameters, such as data rate, control messages (interests and reinforces) periods, and packet sizes, to correspond to the Multi parameters. The simulation results are presented in Fig. 7.

As we can observe, Push and 1PP Diffusion behaviors are close to the SID and EF-Tree, respectively (see Fig. 5). It is due to their similar characteristics of infrastructure construction and maintenance. The main difference in the results exists between SID and Push Diffusion. In Push, exploratory packets are sent in fixed periods, while in SID data are sent by broadcast until paths are built. This difference makes SID more appropriated for event-driven scenarios and Push Diffusion more susceptible to eventual packet losses. In fact, it was omitted but Push solution presented a 10% lower delivery ratio than the others. Obviously, less delivery packets results in lower absolute energy consumption, which is not an advantage in this case. A wider comparison among the related protocols can be found in [19]. In summary, we can see that Multi’s advantage is also observed against these classical algorithms by adapting to variable conditions.

Regarding the previously hybrid adaptive solution presented in [8], it uses a threshold on the number of sources as an adaptation model. This model does not represent the event occurrence characteristic properly. For example, occasional distribution of event detections can lead to a concentrated measurement higher than the established threshold. This can lead to a proactive behavior that may not necessary correspond to an event occurrence characteristic. We also evaluated this solution in the uniform detection scenario, which better presents the variance in detections. We used thresholds of 1 (Multi-1) and 3 (Multi-3), because they leded to the best performance in the lower and high traffic cases, respectively. As we can see in Fig. 8, Multi results with the new adaptive model are closer to the best performance of the others, thus it is a more adaptable solution to respond to event occurrence characteristics.

V. CONCLUSION AND FUTURE WORK

In this work, we applied an adaptive hybrid approach for routing in event-driven WSNs. In particular, we presented and evaluated Multi that unites the behavior of feasible reactive and proactive algorithms for WSNs and adapts autonomously according to the traffic characteristic captured by an event-detection estimation model. Its evaluation showed energy savings in varying network conditions. Thus, when operational conditions of the network are unpredictable, adaptive approaches like Multi should be more suitable than a single strategy algorithm, extending its applicability.

Our future work includes the evolution of Multi. We intend to improve both reactive and proactive algorithms, to evaluate other techniques for event estimation and to include and evaluate other functions like data aggregation and node scheduling.
With the applied approach, it is possible the construction of new hybrid adaptive solutions including other routing algorithms, like those with geographical and hierarchical characteristics. The objective is to find other application requirements and network metrics in which an adaptive behavior among different routing strategies can improve performance.

REFERENCES


