# Multi: A Hybrid Adaptive Dissemination Protocol for Wireless Sensor Networks*

Carlos M. S. Figueiredo[1,2], Eduardo F. Nakamura[1,2], and Antonio A.F. Loureiro[1]

[1] Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, MG, Brazil
{mauricio, nakamura, loureiro}@dcc.ufmg.br
[2] FUCAPI – Analysis, Research and Technological Innovation Center
Manaus, AM, Brazil
{mauricio.figueiredo, eduardo.nakamura}@fucapi.br

**Abstract.** Data dissemination (routing) is a basic function in wireless sensor networks. Dissemination algorithms for those networks depend on the characteristics of the applications and, consequently, there is no self-contained algorithm appropriate for every case. However, there are scenarios where the behavior of the network may vary a lot, such as an event-driven application, favoring different algorithms at different instants. Thus, this work proposes a new hybrid and adaptive algorithm for data dissemination, called Multi, that adapts its behavior autonomously in response to the variation of the network conditions. Multi is based on two algorithms for data dissemination that are also presented and evaluated in this work: SID (Source-Initiated Dissemination), a reactive algorithm where dissemination is started by the source nodes, and EF-Tree (Earliest-First Tree), an algorithm that builds and maintains a tree, in a proactive fashion, to disseminate data towards the sink.

## 1 Introduction

A wireless sensor network (WSN) [1, 2] is comprised of a large number of sensor nodes forming an *ad hoc* network to monitor an area of interest. This type of network has become popular in the scientific community due to its applicability that includes several areas, such as environmental, medical, industrial, and military.

WSNs diverge from traditional networks in many aspects. Usually, these networks have a large number of nodes that have strong constraints like power restrictions and limited computational capacity. In general, WSNs demand self-organizing features, i.e., the ability of autonomously adapt to the eventual changes resulted from external interventions, such as topological changes (due to failures, mobility or node inclusion), reaction to an event detected, or due to some request performed by an external entity.

The objective of such network is to collect data from the environment and send it to be processed and evaluated by an external entity through a sink node. Consequently, data dissemination towards the sink node is a fundamental task and, depending on the application, it can be performed considering different models [3]. In a continuous monitoring, a monitoring application receives continuously sensing data from the environment. In an event-driven monitoring, a sensor node sends a notification to the monitoring node when an event of interest happens. In an observer-initiated

* This work was partially supported by CNPq, Brazilian Research Council, under process 55 2111/02-3.

monitoring, an observer sends a request to the network that gathers data about the sensing data and sends it back to the monitoring node. In a hybrid strategy, a WSN can also allow different types of data delivery, like in a scenario that reports the gathered data whenever the observer requests it, and the sensor node immediately sends a report of a critical event whenever it happens.

Different algorithms [4–7] have been proposed to disseminate (routing) data gathered by sensors. However, different applications and scenarios demand algorithms with different features. This will be shown in this work through the presentation and analysis of two algorithms: SID (Source-Initiated Dissemination), a reactive algorithm for event driven scenarios; and EF-Tree (Earlier-First Tree), a proactive algorithm for scenarios with intense data communication.

Thus, given a specific scenario, the WSN can be designed to operate with the most appropriated data dissemination algorithm, which can be defined *a priori*. However, in some cases the variations of these scenarios can be constant or even unpredictable. For instance, an event-driven scenario may present a low incidence of events and, at a given moment several events can be detected generating a high traffic. For those situations, it should be provided algorithms, which are appropriated for different periods, such as SID and EF-Tree, and it might be infeasible or undesirable to an external entity to act dynamically on the network to change its behavior.

Ideally, the network should be able to self-organize itself according to such variations, adjusting its behavior to assure that all functions (e.g., sensing, communication and collaboration) will be energy-efficient during its lifetime. This work presents and evaluates a hybrid algorithm that adapts itself in a scenario with variations of the traffic load. The algorithm, called Multi, aggregates the features of both, SID and EF-Tree, and autonomously alternates its operation to fit the current condition. This approach represents a new strategy to build data dissemination algorithms for WSNs.

The remainding of this paper is organized as follows. Sections 2 and 3 describe the data dissemination algorithms EF-Tree and SID, respectively. Section 4 presents Multi, an adaptive algorithm based on EF-Tree and SID. The evaluation of the algorithms is presented in Section 5. Section 6 compares our solution to other algorithms presented in the literature. Finally, Section 7 presents our final considerations and discusses some future directions.

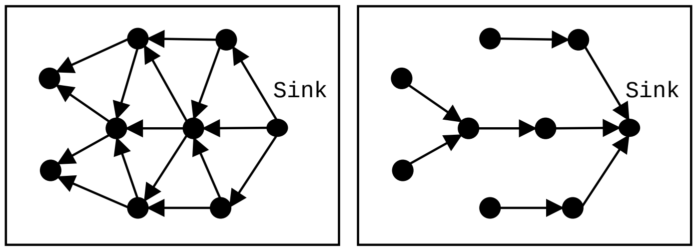
## 2 EF-Tree (Earlier-Firt Tree)

Usually, WSNs collect data from the environment delivering it to a more powerful system that stores, analyzes and deliveries such data to the users. Generally, there is an special element, called sink node, responsible for connecting the WSN to an infra-structured environment. Thus, data collected by the sensor nodes must be disseminated towards a sink node, usually in a multi-hop fashion.

A simple and efficient structure for data dissemination is a routing tree [8]. This structure is created and maintained by a sink node in a proactive fashion, connecting all reachable nodes. Our implementation, called EF-Tree (Earlier-First Tree), has the particularity of rebuilding periodically the tree to cope with eventual topological changes. The building process works as follows:

- The sink node starts the process by broadcasting a control message as depicted in Fig. 1(a).
- When a node initially receives the building message, it identifies the sender as its parent and broadcasts the building message. Messages received from other neighbors are discarded. Note that it is possible to define other possibilities to choose a parent node such as to choose the node that belongs to a path with the highest amount of energy available.

– Whenever a node has a data to be transmitted (sensed or forwarded by another node), it will send it directly to its parent (Fig. 1(b)).
– The building process is periodically repeated so the network reflects eventual topological changes, such as failures, node movements and inclusion of new nodes. This periodicity depends on the frequency topological chances occur. More dynamic networks need shorter rebuilding periods.




(a) Tree building. (b) Data dissemination towards the sink.

**Fig. 1.** EF-Tree algorithm.

## 3 SID (Source-Initiated Dissemination)

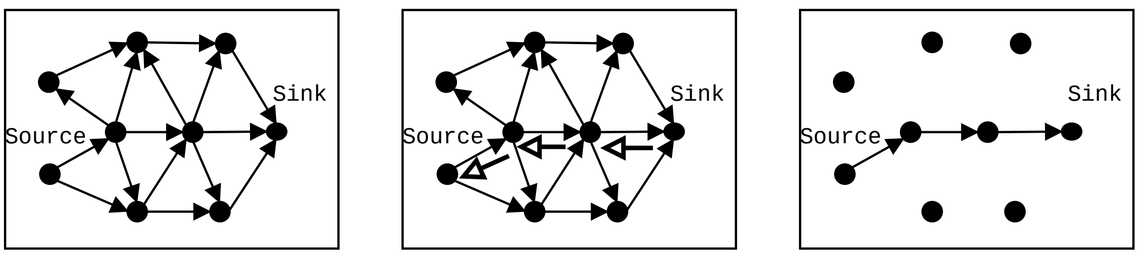
SID is a reactive algorithm for data dissemination. In this case, the process of creating the data dissemination infra-structure is initiated by the source node whenever necessary (e.g., when an event is detected). This principle is appropriate for applications that disseminate data whenever an event is detected.

For event-driven scenarios, this approach is better than a proactive algorithm, such as EF-Tree, because it does not need to continuously maintain an infra-structure for data dissemination. This is particularly true in scenarios where events have a low occurrence rate (hours or days).

In this approach we assume that sensors are already deployed and pre-configured to detect an event of interest, in the same way a sprinkler is already configured to detect fire. The network will remain inactive until events are detected and the communication process is started by sensors that detected them. The algorithm works as follows:

– Nodes that detect an event broadcast their data, the node's identification and a timestamp (Fig. 2(a)). This allows the identification of the data.
– Whenever a node receives the data sent by another node, it stores its identification (source identification and timestamp), as well as the sender identification. As the disseminated data is broadcasted, the node will receive it from all neighbors, however, it will register and forward only once (the first data received). Like the EF-Tree, another criteria can be considered.
– Similarly, data will arrive at the sink node from all of its neighbors. Then, the sink will send a control message requesting the data to be sent by the node from which it firstly received the data. This message identifies the data to be sent.

- When a node receives this control message, it repeats this process identifying which node should send that data to it. This process is repeated until the source nodes are reached (Fig. 2(b)).
- Once a source node receives the control message requesting its data, it will update its own table so subsequent data will be sent to the node that firstly requested its data. Thus, data will be disseminated through the path where the sink's requisition message arrived, which might be the fastest path (Fig. 2(c)).
- In order to allow the network to adjust to eventual topological changes (due to failures, mobility or node inclusion), 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, the node 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.
- Once the events disappear, data will not be generated anymore and, consequently, the sink node will stop sending requisition messages to the sources. In absence of the periodical requisition messages of a specific data, the table entries will expire and the network will become inactive again.




(a) Data diffusion. (b) Requisition messages. (c) Data dissemination towards the sink.

**Fig. 2.** SID algorithm.

The idea of requesting data periodically, finding paths between sources and a sink node is interesting mainly when an event occurs and does not stop immediately. The frequency of requesting data should be lower than the data generation avoiding the overhead of flooding every data and the overhead of sending a requisition message for each data.

## 4 Multi: A Hybrid Adaptive Protocol for Data Dissemination

Clearly, in the algorithms described above (SID and EF-Tree), we can observe that they are appropriate for different scenarios. Comparatively, SID should used whenever the event ratio is low, so there is no need to constantly maintain a network infra-structure such as EF-Tree does. With SID the infra-structure is created and maintained on demand, i.e., only when there is any data to be transmitted.

When the communication load is high, such as in scenarios with continuous sensing and many sources, SID is not an appropriate choice because it maintains a dissemination scheme for each source, which is started by a flooding causing a waste of energy and bandwidth. In such cases, EF-Tree is clearly more appropriate.

As previously mentioned, in an event-driven scenario, the events might occur with a non-uniform distribution. For instance, the network may remain with a very low activity for days, which is good for SID, however, in a given moment several events might occur generating a traffic large enough to use the EF-Tree. Choosing an algorithm a priori surely will not achieve the best results during the entire network lifetime. Thus, we propose a hybrid and adaptive algorithm, called Multi, that incorporates the features of both algorithms adapting its behavior according to the current network condition. This approach, to the best of our knowledge, is new in the WSN domain. Furthermore, in this work we propose a new class of algorithms for self-organization.

The current implementation of Multi incorporates the characteristics of SID and EF-Tree. The adaptive control of Multi will be performed by the sink node that monitors the amount of events detected by the network, since both EF-Tree and SID control schemes are performed by the sink node. The operation of Multi obeys the following steps:

- Initially, the network operates just as SID. When a node detects an event it floods the data in the network. Once the sink receives the data, it sends requisition messages towards the sources, so the paths between each source and the sink is formed. These requisitions are sent periodically while the sources keep sending data.
- The sink node computes the amount of sources sending data in a predefined period. In our implementation, we made this period to be the same of the requisition period.
- Once the amount of sources exceeds a given threshold, the sink node begins to send messages to build the tree, just as the EF-Tree does. This comes from the observation that for a given number of sources, we can wait its increase, and it is less costly to build and maintain a dissemination infra-structure to the entire network than for every possible source.
- When a node receives a building tree message, it keeps the parent's information for a validation time, which is greater than or equal to the periodicity of the building messages. After this, if the node has a valid parent it will always send its data to the current parent.
- If the event rate diminishes to a value lower than the predefined threshold, the sink node will stop building the tree. Consequently, the parent of each node will become invalid (the validation period will expire) so the network will operate again as SID.

The definition of the threshold value for changing the routing algorithm depends on several factors that impact the execution cost of each algorithm. For instance, the network size, the amount of sources (traffic), traffic duration and data rate. Thus, this threshold should be dynamically computed based on models of event occurrence and the previous behavior. However, for the sake of simplicity, it will be defined a static threshold based on the simulations of SID and EF-Tree algorithms.

## 5 Simulation and Evaluation

In this section, the algorithms previously described are evaluated through simulation. Initially, we will describe some particular scenarios assessing the performance of SID, EF-Tree, Flooding and Directed Diffusion. Afterwards, we will show the adaptive ability of Multi and its advantages compared to SID and EF-Tree alone.

The experiments were performed using the ns-2 Network Simulator [9]. The simulation parameters were based on the Mica2 Sensor Node [10] using the 802.11 protocol in the MAC layer (see

Table 1). In all simulations, we considered only one sink, data messages of 20 bytes were transmitted every 10s, and control messages of 16 bytes were transmitted every 100s.

| Parameter | Value |
|---|---|
| Transmission power | 33,3mW |
| Reception power | 30,0mW |
| Bandwidth | 76800 bps |
| Communication radius | 40m |

**Table 1.** Mica2 parameters used in the simulations.

### 5.1 SID versus EF-Tree

To compare the behavior of SID and EF-Tree we defined two scenarios. In the first one, data is disseminated continuously, where several nodes send their data periodically towards the sink. In this scenario we will evaluate the scalability of the algorithms according to the size of the network, and their resilience in presence of failures. In the second scenario, events occur randomly with random duration. The metrics evaluated were the delivery ratio to assess failures, and energy consumption, the most restrictive resource in a WSN.

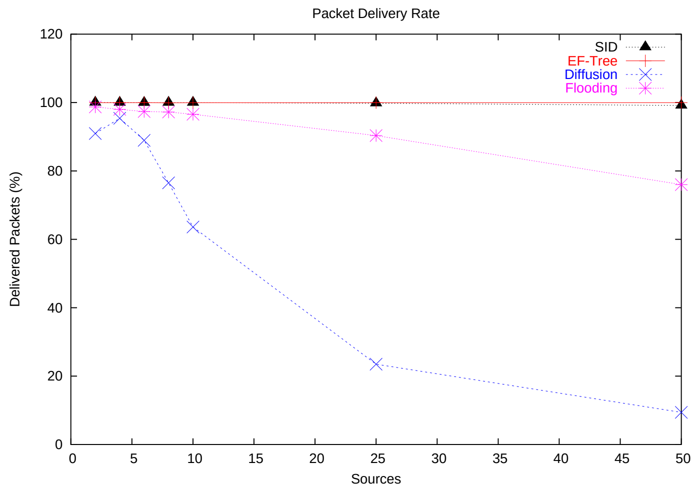
Both algorithms are also compared to both the classical flooding approach and the Directed Diffusion algorithm. For the Directed Diffusion, interests were disseminated every 100s (equal to the building period of EF-Tree and data request in SID), interests were valid for 200s and negative reinforcement was enabled.

Figure 3 shows the behavior of the algorithms in a scenario with 50 nodes distributed uniformly over a $100 \times 100m^2$ area, with continuous traffic. The number of sources varied from 2 to 50 nodes, randomly chosen.

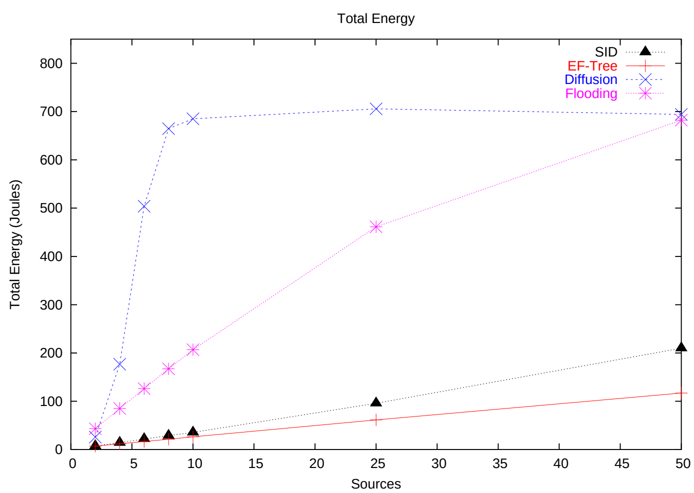
As we can see in Fig. 3, flooding a data impacts the delivery rate (Fig. 3(a)) on a network with bandwidth constraints. This can be observed in the Flooding algorithm with the decrease of delivered packets when the number of sources increases. The Directed Diffusion, which floods interests and data to setup gradients, we can observe a more drastic impact given by the loss of control messages, because paths are not reinforced so broadcasts are maintained. This occurs because the network cannot process the data rapidly, overfilling the queues of routing nodes, which results in losses. SID also starts the dissemination process by flooding the data, however, once a path connecting sources to a sink are established, no data is flooded anymore and the queues of routing nodes are not overflowed, keeping high delivery rates such as EF-Tree.

Regarding the energy consumption, EF-Tree presented the best result (Fig. 3(b)), showing the advantage to build proactively the dissemination infra-structure to the network in this scenario. SID has a higher increase on energy consumption because of the need to build this infra-structure for every source. In addition, Directed Diffusion consumes more energy than EF-Tree and SID, and less than Flooding with two sources, and operates saturated with more than ten sources. This poor performance is consequence of the impact of the delivery rate, as described before, and shows that it is not viable for the simulated scenario (low bandwidth and high number of sources).

The scalability of the algorithms was evaluated in the previous scenario, however, the number of sources was fixed (20 nodes) and the network size varied from 50 to 200 nodes. As we can see




(a) Packet delivery ratio.

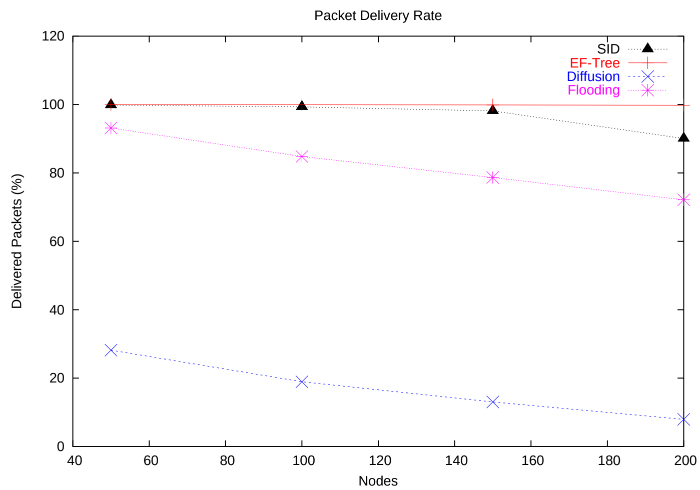



(b) Absolute energy consumption.

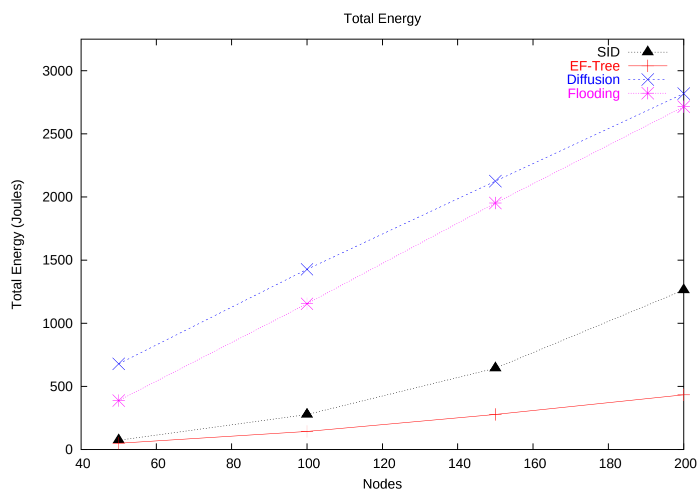
**Fig. 3.** Continuous data traffic in a 50-node network.

in Fig. 4, the impact of the network size is greater in the algorithms based on flooding, because of packet losses, as described before (Fig. 4(a)). Clearly, in this scenario, EF-Tree also scales better than the others when we evaluate the energy consumption (Fig. 4(b)). SID spent more energy than EF-Tree due to the initial flooding of data and, again, Directed Diffusion presented a worse behavior than SID and EF-Tree, working saturated.

To evaluate the resilience of the algorithms, we fixed the number of sources in 20 nodes and varied the probability of failures from 0 to 50% randomly during the simulation. When a node fails, it stays inactive until the simulation ends. As we can see in Fig. 5, SID and EF-Tree outperform




(a) Packet delivery ratio.




(b) Absolute energy consumption.

**Fig. 4.** Scalability.

the others when the failure probability is low. Their delivery rate decreases and stays close to 90% with the increase of failures. Flooding maintains its performance close to 90%, with losses related to high broadcasts. The Directed Diffusion has low delivery rate due to the saturated operation as in the previous simulations.

Now we turn to the second scenario, based on the previous one that represents an event-driven application. To represent the random occurrence of events, sources generated data randomly along the simulation time. The duration of data generation was a random value between 1 and 50s.

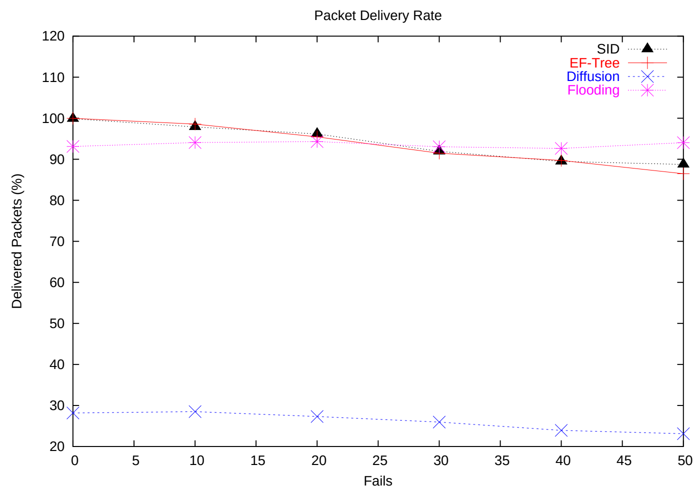


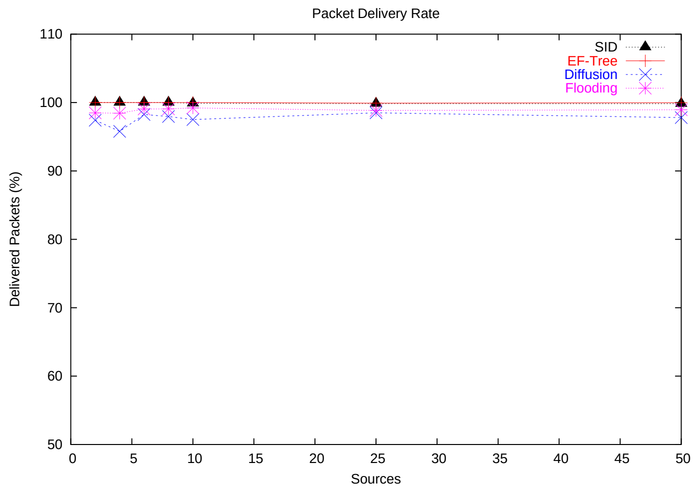

**Fig. 5.** Packet delivery ration under different probabilities of failure.

As we can see in Fig. 6(b), SID outperforms the other algorithms when the number of sources is low. This occurs because the dissemination infra-structure created by SID tries to use only the necessary nodes connecting the sources to the sink node. Besides, this infra-structure is created and maintained only when necessary. However, when the traffic increases, EF-Tree begins to be more adequate because it builds the dissemination infra-structure for all nodes at once. We can also observe that when the traffic is not very intense, all algorithms present better packet delivery ratio (Fig. 6(a)). Comparing to the previous scenario, Directed Diffusion improved its performance, which reinforces the fact that it is appropriate to scenarios with few sources only.

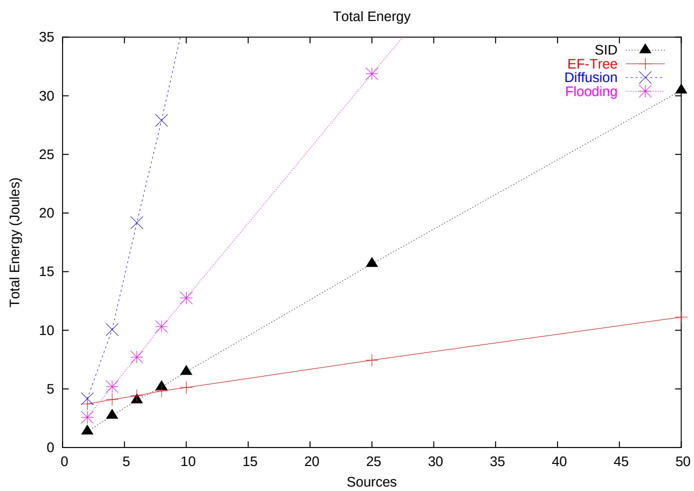
### 5.2 MULTI

To illustrate the adaptability of Multi, we created a scenario where the occurrence of events varies along the simulation time. Basically, we generated a wave of events in the network, as it occurs when a mobile target is detected. At the instant the target begins to invade the sensor field few nodes are able to detect its presence. When the target is fully present at the sensor field the number of sensors that detects it is maximized. When it begins to leave the sensor filed, again, few nodes report its presence, and when the target is gone, the network becomes inactive again. We configured Multi to change the routing strategy when the threshold of three sources is identified. This value was chosen based on the previous simulations that showed that SID outperforms EF-Tree for scenarios of few sources (less than six sources).

In Fig. 7 each point of the curves represents the energy consumption accumulated in the last 10 seconds. In these simulations, data was generated from the instant 500 to 900s. Fig. 7(a) shows that while the number of sources does not reach the threshold, Multi behaves exactly as SID outperforming EF-Tree in the intervals of inactivity (0 to 500s and 900 to 1400s), because EF-Tree keeps building the tree even when no data is generated (energy increases at every 100s). At the beginning of the activity period (about 500s) the consumption of SID and Multi started to increase




(a) Packet delivery ratio.

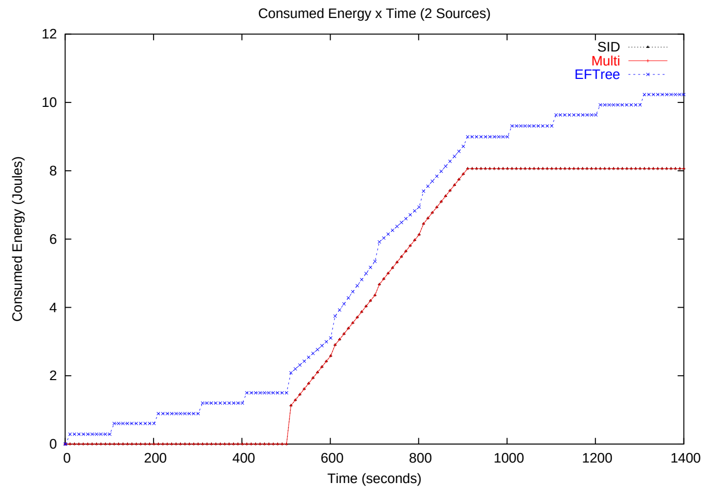



(b) Absolute energy comsumption (Ampliated).

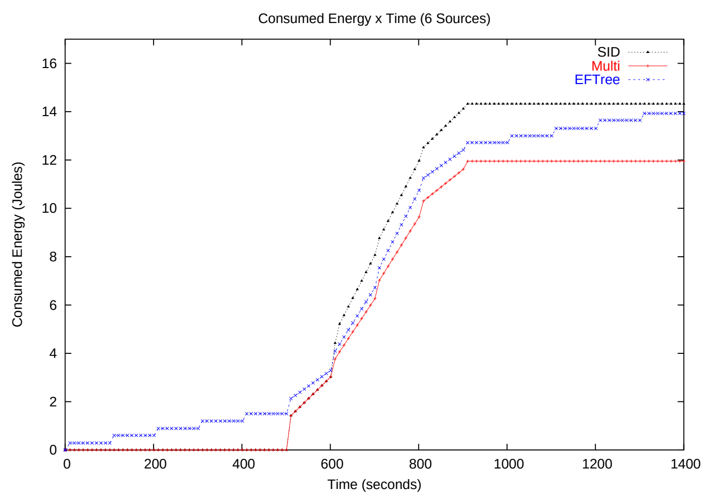
**Fig. 6.** Random events of random duration.

because of the initial data flooding started by the sources. Shortly after, Multi consumption is similar to EF-Tree because the paths connecting sources to sink are established.

When the number of sources increases, we noticed that SID consumed more energy compared to EF-Tree during the period of data traffic (Fig. 7(b)). This occurs because of the larger number of nodes flooding its data (instants 500s and 600s in Fig. 7(b)). At this point, Multi adapts its behavior as soon as some sources begin to generate data starting to operate as EF-Tree and outperforming SID. When the traffic reduces again (instants 800 and 900s in Fig. 7(b)), Multi changes it behavior again to operate like SID. This adaptation allows to save energy compared to SID and EF-Tree.




(a) Two sources.

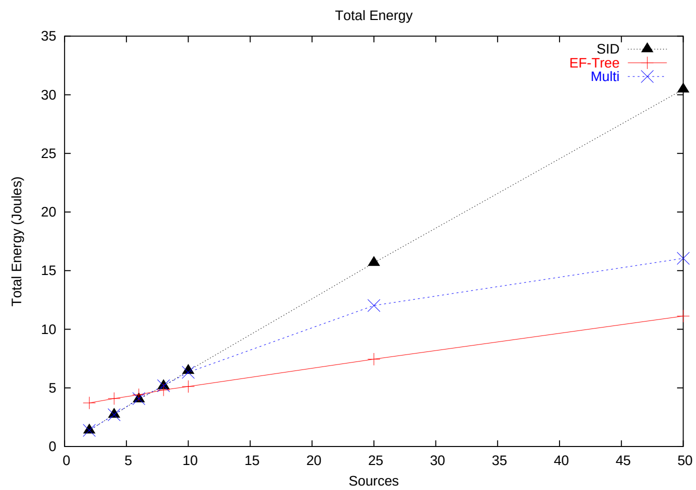



(b) Six sources.

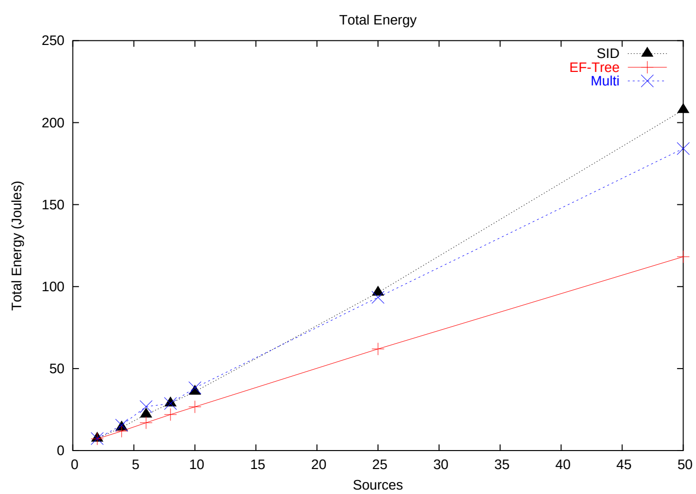
**Fig. 7.** Traffic based on a wave of events.

We also evaluated the behavior of Multi under the scenarios of exhaustive simulation previously used to evaluate SID and EF-Tree. The energy consumption of each algorithm at the two different scenarios is depicted in Fig. 8.

The results showed that Multi outperforms EF-Tree when the number of sources is low because in such cases the algorithm behaves as SID. When the number of sources increases, Multi starts to behave as EF-Tree outperforming SID, which is not suitable for such scenario (Fig. 8(a)). This happens because the simulation time was kept constant and the increase in the number of sources leads to a decrease in the inactivity time amortizing the cost of the EF-Tree. In a real event-driven




(a) Energy consumption in the event-based scenario.




(b) Energy consumption in the continuous scenario.

**Fig. 8.** Multi versus SID versus EF-Tree.

scenario, we can expect to have longer inactivity periods than in the simulated scenarios. Thus, the energy cost of the EF-Tree is shifted to a higher value due to its proactive characteristic what results in a better performance of both SID and Multi.

After the transition points, we can observe that the energy cost of Multi is not the same of EF-Tree that happens because Multi always starts to operate as SID changing to EF-Tree only when the traffic threshold is reached. The difference observed at the transition points depends on the conditions of the simulated scenarios (simulation time, data rate and network size). In the scenario of continuous traffic (Fig. 8(b)), Multi does not present great advantages compared to SID because all traffic was generated in the beginning of the simulation, so Multi adapted only when the greater communication cost had already occurred. An advantage can be observed in Multi, though, compared to SID when the number of sources is more than 25, and it happens because the energy cost of building a tree is less than the cost of sending individual messages to a large number of sources.

In an event-driven scenario, it is worth to mention that the more sporadic the event is, the greater the advantages of using Multi rather than SID or EF-Tree. On the other hand, in a scenario of intense traffic, it might be preferable to use only EF-Tree.

## 6 Related Work

Several routing algorithms have been proposed to solve the data dissemination problem in WSNs [4–7]. However, none of these algorithms presents a definitive solution for different scenarios. In fact each algorithm is designed do operate in a specific scenario and application. Directed Diffusion [4] was the first algorithm for data-centric routing in WSNs and, although, it is able to adapt to eventual topological modifications it is not the most appropriate solution for event-driven scenarios because it needs to flood the interests periodically, which results in waste of resources (specially when the number of sources is large). There are few data dissemination algorithms designed for event-driven scenarios. An example is SPIN [7] that tries to suppress the weakness of flooding by local negotiations (that use data identifications only) so data is sent only for nodes that are interested in them.

Recently, it was proposed another work, related to the one presented in this work, that deals with the necessity for matching the data dissemination algorithms to the application characteristic [11]. In that work, two algorithms, extended from Directed Diffusion, are proposed: Push Diffusion, for few sources; and One-Phase Pull Diffusion, for a high number of sources. The characteristics of those algorithms are similar to SID and EF-Tree. However, this work consists of a new approach to design data dissemination algorithms exemplified by Multi, which incorporates the features of two other algorithms adapting itself automatically when certain communication conditions are verified. Also, a more realistic evaluation was done by the simulation with real sensors network parameters (based on Mica2), while the work in [11] used an ideal scenario counting only the number of transmitted messages.

Other efforts related to the design of adaptive and hybrid algorithms can be found in the literature. For example, SHARP [12] is a routing protocol for ad hoc networks that founds an equilibrium between reactive and proactive protocols adapting the degree of how routing information is propagated in the network. This work is different by focusing on data dissemination characteristics of WSNs.

## 7 Conclusions and Future Work

In this work we presented a new approach for self-organizing wireless sensor networks to allow the autonomous adaptation of the data dissemination strategy to improve the energy efficiency when the scenario presents different variations on its traffic condition. This was done through the presentation and evaluation of two different algorithms for data dissemination in WSNs, SID and EF-Tree, showing their behavior in different scenarios, and introducing Multi, that unites the behavior of both algorithms so it can adapt itself according to the number of events that are detected by the sensor nodes.

It is true that in WSNs each algorithm is appropriate for a different application and a different scenario. However, in scenarios that present high variability on the incidence of events (or data traffic), only one algorithm might not achieve the best result along the network lifetime. When operational conditions of the network are unpredictable, adaptive hybrids algorithms like Multi should be more adequate than a single strategy-algorithm, extending its applicability.

Although the current implementation and evaluation of Multi did not explored its full potential, it clearly provides benefits in some situations, such as the the ones illustrated here. Thus, to make it more generic, its adaptive capability should depend on event occurrence models and prediction models based on the previous behavior.

Future work includes the evolution of Multi, introducing data aggregation functions and adoption of different behaviors for different partitions of the network, according to the model of event occurrence. Also, we plan to design new hybrid adaptive algorithms based on the approach presented here, however, including other data dissemination algorithms such as Push Diffusion and One-Phase Pull Diffusion briefly mentioned above.

## References


1. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. Communications of the ACM **43** (2000) 51–58
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: Wireless sensor networks: A survey. Computer Networks **38** (2002) 393–422
3. Tilak, S., Abu-Ghazaleh, N.B., Heinzelman, W.: A taxonomy of wireless micro-sensor network models. ACM Mobile Computing and Communications Review (MC2R) **6** (2002) 28–36
4. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, USA, ACM Press (2000) 56–67
5. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocols for wireless microsensor networks. In: Proceedings of the Hawaiian International Conference on Systems Science (HICSS), Maui, Hawaii, USA (2000)
6. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review **5** (2001) 11–25
7. Kulik, J., Heinzelman, W., Balakrishnan, H.: Negotiation-based protocols for disseminating information in wireless sensor networks. Wireless Networks **8** (2002) 169–185
8. Sohrabi, K., Gao, J., Ailawadhi, V., Pottie, G.: Protocols for self-organization of a wireless sensor network. IEEE Personal Communications **7** (2000) 16–27
9. NS-2: The network simulator - ns-2 (2004) [Online] Available: http://www.isi.edu/nsnam/ns/.
10. Crossbow: Mica2 - wireless measurement system (2004) [Online] Available: http://www.xbow.com/Products/New_product_overview.htm.
11. Heidemann, J., Silva, F., Estrin, D.: Matching data dissemination algorithms to application requirements. First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003) (2003)
12. Ramasubramanian, V., Haas, Z., Sirer, E.: SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. In: Proceedings of the 4th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03). (2003) 303–314