Energy efficient multi-sink positioning and architecture for Wireless Sensor Networks

Karl Baumgartner and Stephan Robert
University of Applied Sciences of Western Switzerland, Yverdon

Abstract— **Pollution is nowadays emerging as a major threat for the human being. Cities' sizes are increasing, as well as the number of inhabitants living in urban areas. Therefore, air pollution monitoring becomes an important concern for the scientific, medical and political world. This paper describes a scalable architecture for monitoring air pollution in large areas where collected data will be represented through an Internet interface. The proposed solution is based on the presence of multiple sinks in the Wireless Sensor Network (WSN). An algorithm to optimize the placement of sinks is presented.**

***Index Terms*— Wireless sensor network, Ad-hoc network, Cluster-based routing, Pollution monitoring, Sinks positioning**

I. Introduction

Environmental monitoring has generated a large amount of information for scientists. The purpose of these experiences is to improve our knowledge about the impact on nature of human being. Data are collected by sensors, which are usually measuring temperature, humidity or air pressure. However, any sort of gas may be monitored for the need of a specific project. Fifteen years ago, the only suitable way to centralize information coming from different sensors was the use of cables. This was an important barrier to the development of sensor networks. Nowadays, Wireless Sensor Networks (WSN) have permitted new ways of research. Nodes can be organized on an ad-hoc wireless architecture (no more fixed infrastructures). Sensor networks often have a root point called sink where data are collected and sent via a long range radio system (GPRS, UMTS, WiMAX, …). The radio coverage of each node is usually not large enough or too expensive to reach a sink. Hence, the nodes have implemented a protocol called Multi-hop routing to allow them to send information to the sink via other nodes [1]. The large size of our sensor networks forces us to employ multiple sinks.

This article describes and discusses architectures for implementing a solution to centralize data from multiple sinks. We propose a solution and practical tools which collects the sensor measurements and dispatches them through a TCP/IP-based network. The node sink is connected to an embedded computer via the RS-232 port where the data are temporary stored inside a database. Then, data saved in every sink are sent to the centralized database server. The position of the sinks influences the energy spent by each node for communication and consequently the expected lifetime of the sensor network. Thus, we have elaborated two methods to choose the most appropriate position of the basestation. Additionally, the simulations made with Matlab confirm the validity of our algorithms.

In the first part of this article, we describe the usual constraints we face with Wireless Sensor Networks. The second part concerns a description of the architecture of how data are collected. Then, positioning algorithms of the sinks are presented in the third part of this article. Finally, the last part summarizes the existing work related to ours.

II. Wireless Sensor Network

A new generation of low-cost wireless communication nodes has emerged [2]. Most of them are implementing TinyOS, which is an open-source operating system developed by the University of California at Berkeley.

A. *Multi-Hop Routing*

The data generated by a sensor have to be gathered to a basestation node commonly called *sink*. Usually, the sink is out of the radio coverage range of many nodes. Therefore, a multi-hop routing protocol is designed to allow every node to route the packet to the destination node (sink). A kind of gathering tree is set up as described in Figure 1. This type of architecture is called *cluster-based routing* and has already been discussed in the literature [3] [4] [5]. It is important to note that each node only has information about its parent address. Nodes may also reduce the amount of information sent by executing data aggregation [6] [7] [8]. Often, cluster-based protocols are implemented to work with only one sink in the whole network. However, if the number of node is significantly large, the nodes directly connected to the sink will rapidly run out of battery. This is due to the fact that they will route every packet originated from a large number of nodes behind them on the tree. Therefore, a multi-sink router protocol is the unique way to solve this issue [9] [10].

This paper is not describing and discussing about how to implement such an algorithm. We consider more particularly the way for collecting data from the multiple sink to the centralized database. We also underline a few important points to consider about the wireless sensor network.

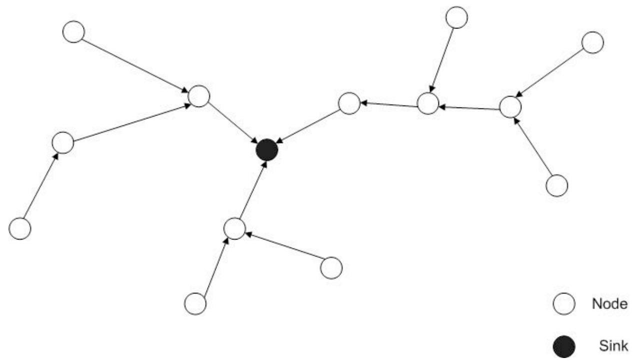

Figure 1: Multi-Hop connection scheme which allows all nodes to send data to the root node, called sink

B. Hardware

The most important element of the project infrastructure is the nodes which should implement the wireless communication to the sensors. We want to use a materiel which supports TinyOS and which is low power based. Tinynode 584 (Shockfish) with its extension board (Figure 2) matches our needs [14].

Figure 2: Tinynode 584 and its Standard Extension Board: platform for sensor networking

The microcontroller is an MSP430F1611 (Texas Instrument) with 10 kB of RAM, 48 kB of flash and 128 B of memory. The CPU is a low power 16-bit RISC with 16 bit register. Wireless transceiver: Xemic XE1205 with the following key features: RF output power up to 15 dBm (programmable), high sensibility down to -121 dBm at 1.2 kbit/s, bit rate up to 152.3 kbit/s, continuous phase 2-level modulation, operate in the 433, 868 and 915 MHz (but tinynode 584 only use the 868), low-consumption and many other useful features as RSSI (Received Signal Strength Indicator). Furthermore, this material is optimized to run TinyOS and is based on an ultra low power 3 V design. The interface RS-232 and Jtag (principally used for debugging) are present on the extension board.

III. Architecture

A. Project overview

The architecture of this sensor network is described in Figure 3. Every sink is connected to Internet, which allows them to transmit the sensor information to a centralized database server. For this purpose, several different ways to do it are possible. It slightly depends on the geographical location of the sink and the physical available communication infrastructure. Four standard access technologies have been used: GPRS, ADSL, Analogue Modem or direct connection to the Internet. For all these solutions, the TCP/IP stack has to be implemented. Consequently, an embedded PC implementing the TCP/IP stack will be part of the sink. Also, the data will have to be buffered inside this computer in case of failure or disconnection of the TCP/IP network. For this reason, a database (Mysql) has been used. Finally, the port RS232 will interface between the embedded computer and the node. The programming language which has been used is Java, principally because of its portability. Furthermore a large number of APIs are available, which reduces the development time. Globally, the sinks collect the data coming from the ad-hoc network and store them into local databases. Periodically, data are sent to the centralized database server through the TCP/IP stack. During the whole process the integrity of data is guaranteed.

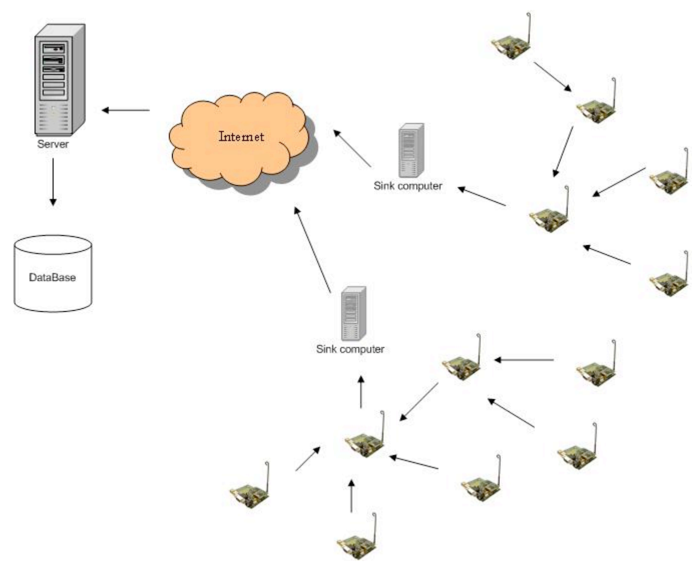

Figure 3: Architecture of data acquisition of the WSN using multiple sink and a centralized database

B. Detailed architecture of the software

Data coming from the sensor are routed to the sink node via the multi-hop routing protocol. Then, the sink node will transit the messages through the RS-232 interface to an embedded computer called sink computer (the sink node and the sink computer are geographically at the same location). The sink computer has the following main logical functionality: MySQL database, TCP/IP stack and a Java virtual machine.

The MySQL database is essential because the TCP/IP connection is not hundred percent reliable. Consequently, data should be saved locally on the sink computer and should be sent as soon as the TCP/IP connection is restored. The database server collects the information coming from the multiple sinks. It is important to note that it is always the designated sinks that will set up the connection to the server. This is because the sinks will probably be hidden behind a Network Address Translation (NAT) server which shares a single IP address to many end users.

Then, the General Manager is the tool used to control and manage the totality of the architecture. This application will be a kind of log centre. Every log message generated by the sink application will be automatically sent to it. Another functionality of General Manager is to control the Wireless Sensor Network where many parameters may be changed. Finally, the server database can be accessed easily and user-friendly from anywhere on the Internet.

1) *Sink computer programs*

The sink computer needs the following components: an MySql database, a TCP/IP stack and a Java virtual machine. Consequently, the chosen programming language used for the implantation has been Java (J2SE 1.4.2). TinyOS, which is an open source operating system, furnishes classes to get the data from the nodes through the RS-232 interface. The communication through the serial port is implemented by the RFC 1662 standard (Point-to-Point Protocol). In fact, the classes interpret this protocol and return the packet sent by the nodes in a readable form; They are used by the class SerialForwarder. These interpreted data are then transmitted to the Exchange program which checks their integrity and store them into the local database. Then, this database is periodically examined by SinktoServer which sends the new data (mean the data not yet transmitted) to the Database server. The connection between the sink and the server is realized through TCP/IP.

2) *Database Server*

The database server is basically a computer with the following characteristics: high hard disk capacity, Mysql database, TCP/IP stack, Java virtual machine and a public internet IP. It is important to note that the server is not able to establish a communication with the sinks. Thus, the program ManageSocket should accept a connection from the sink at any time. Afterwards, the sink transmits the new data which are saved to the centralized database. At the end of this procedure, the server should acknowledge the sink that the data has been well stored. Therefore, the data will be sent only once.

To know if a node or a sink crashed down, a program is periodically checking the database to determine if suddenly a node stops sending information. This work is done by the program Checknode which communicates with the General Manager.

3) *General Manager*

A third important module to allow a good functionality is the General Manager. Its purpose is to centralize the warning and the errors coming from the nodes, the sinks and the database server. Instead of having a simple log file on each sink, the General Manager will acquire all of them. The Java class logger is the perfect tool to do such things. With this class, the log message may be classified in seven levels of severity. For the project, only the three last levels will be used (Severe, Warning and Info). Then, handlers can be added which forward this log to different destinations. For our case, the logs are forwarded to files and also to a socket (through the TCP/IP network). Hence, the General Manager is needed by the system administrator to summarize all the logs.

Another functionality of this module is to remotely control the nodes. To be able to do that, the sink computer should have another program which allows the General Manager to send an order to it. This program interprets this order and transmits messages (readable by the node) to the SerialForwarder (located in the sink computer) which sends it through the RS-232 interface to the node. Actually just a few node options may be changed:

- The nodes may be reinitialized
- The nodes may change the frequency time which they send value
- Reset the timer
- The power of transmission
- Stop sending information
- Other parameter will still be available…

The General Manager is independent of the database. Thus, its location may be the same as the server database but it may also be to another location on the internet network.

IV. Optimal Positioning of the Sinks

During the design phase of a WSN, the positioning of the sink is crucial. The location of the sinks will influence considerably the energy consumption of the whole WSN. The simplest method to find the sink location is to use a well known algorithm to cluster objects like k-mean. This technique will give a good initial result. However, some improvement may be provided in order to increase the life expectation of the WSN. Our sink location will not necessarily improve the overall consumption but the sum of the Euclidian distances between the consumption of each node and the average node consumption. The result of the optimization is that all the nodes will run out of battery at almost the same time. Two different algorithms are described in this section: the Optimized Free Sinks Positioning (OFSP) and the Optimized Nodes Replacing Sinks Positioning (ONRSP). Simulations made with Matlab show the usefulness of these algorithms.

1) Path Loss and Power consumption

The first point about creating a virtual wireless sensor network is to choose a path loss model for the radio signal. This model will influence considerably our simulation because consumption of each node will depend on this parameter. We have chosen the Hata Model [22] because of its suitability for frequencies between 150 MHz and 1500 MHz. The Hata model is a valid formula for the urban area propagation path loss (deducted from the Okumura measurements).

The quantity of energy spent by the node when the module radio is employed is not only the power of transmission. To send a packet, the radio chip has to be activated for transmission; thus the overall consumption is roughly represented by the following formula [23]:

$$P_{tot} = \alpha + \beta \cdot P_{Tx}$$

P_{Tx}= transmission power (find with hata formula)
α=constant power for the use of radio chip
β=constant factor
P_{tot}=Power total for transmitting with radio chip

$$E_{packet} = \frac{N_{bit}}{B_{rate}} \cdot P_{tot}$$

E_{packet}=the energy to send a packet
N_{bit}=Number of bit in a packet
B_{rate}=bit rate

Now, we can find the energy needed to transmit packets from one node to another in this WSN. For the parameter B and A, we deduced them from the Tinynode datasheet.

2) Tree construction

The sensor nodes have to be deployed on the field for the simulation. They may be spread uniformly on the entire surface (for instance for farming application). For pollution monitoring, the sensor node location is difficult to predict for an urban area due to the high density of obstacles. Consequently, the node placement will be randomized for our simulation. Afterwards the node deployment and the sinks locations (which has not been discussed yet), trees (Figure 1) has to be built. To find the closest path in the graph, the Dijkstra [24] algorithm has been used. The weight between nodes is the power consumption.

Figure 4 represents a graph generated by Matlab for 50 nodes and 3 sinks. In this simulation, we have limited the power of transmission of the node to 15 dBm and the reception sensibility to 100 dBm (according to the Tinynode characteristics).

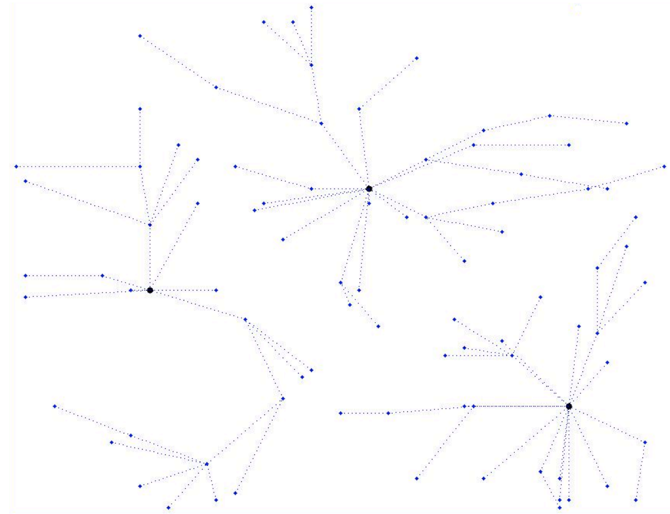

Figure 4: 3 Tree obtained by simulation for three sinks

3) Life expectation of the WSN

The life expectation of a WSN will depend on the fact that all nodes should die approximately at the same time. Statistically, the nodes which die the more rapidly are the one having the largest number of children. Therefore, many death node children may not be able to communicate with the sinks. For that reason, we decided to use a special way to optimize our WSN life expectation. Instead of measuring the WSN overall consumption, the Euclidian distance between the consumption of each node and the average node consumption will be employed. The following formula gives the information for expected life time of the WSN.

$$V = \sum_{n=1}^{N} \left| A_{cons} - C_n \right|$$

V=Sum of the Euclidian distances between the consumption of each node and the average node consumption of the WSN (used to measure the life expectation)
A_{cons}=average consumption of the nodes connected to one sink
C_x=consumption of the nodes x
N=number of nodes in the WSN

Figure 5 represents the zone where nodes will consume the most amount of energy. V is the variable that is going to be optimized. This type of graph can help us to see where nodes will run out of batteries first. A bright color (yellow) indicates that the node consumes more energy than when its color is dark (black)

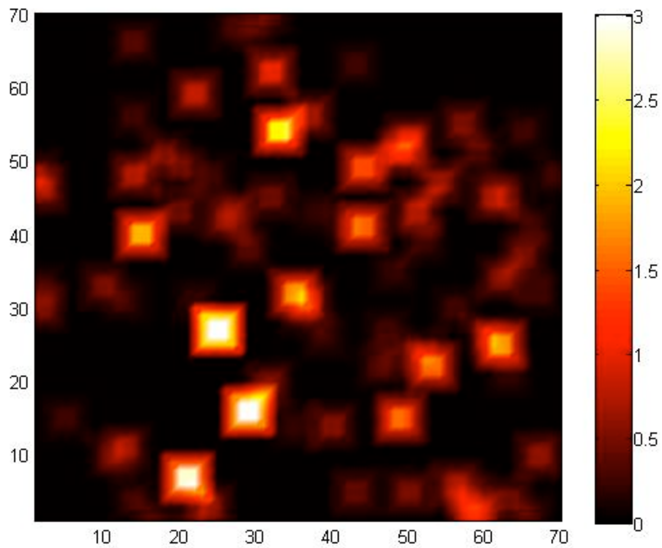

Figure 5: Zone node consumption. A bright color (yellow) indicates that the node consumes more energy than when its color is dark (black)

4) *Initial position of the sinks*

After the node placement, the number of sinks has to be chosen. This number will depend on the architecture and the budget of the WSN. The sinks may be connected to the Internet via GPRS which may have a significantly expensive communication cost. In our algorithm, we consider that the sinks have an infinite amount of energy at disposal, which is rather true in practice (the communication inter nodes is negligible in comparison with the GPRS/UMTS links). Then depending on the environment, we have to decide whether the sink will be located at the same location than the nodes or if the sink may be placed anywhere on the field.

In order to have an initial state for the sink before the optimization, the k-means algorithm [25] gives an initial placement. This algorithm clusters objects based on their weights into k groups. In situation where sinks are placed at a nodes position, the k-means algorithm will find the cluster central point closest to nodes locations. It is important to note that k-means is employed in the literature to discover the final destination of the sink [21].

5) *Optimized free sinks positioning*

After having placed the nodes randomly and the sinks at their initial position using the k-means algorithm explained below, the sinks will be able to move freely anywhere on the field. However, the minimal step movement is 100 meters (it is how the filed was squared). The first move is important in the sense that it will determine the initial direction. The initial move decision is determined by the improvement in term of V (sum of the Euclidian distances). Then, the sink progresses in this direction until no V (Sum of the Euclidian distances between the consumption of each node and the average node consumption of the WSN) improvement is detected anymore. Afterwards, a new direction has to be chosen with the same function as the initial move until improvement is anymore possible. In Figure 6, the pseudo-code of this procedure is written.

```
Find the initial Sink Position with k-means
For j= 1 to NumberOfSink
  Find the best initial direction to follow for Sink j
End For
m=0
Repeat until none improvement of V possible
  m=m+1
  For i= 1 to NumberOfSink
    If none direction selected
      Break the loop For
    End if
    VInit=V with actual sinks
    PositionInit=sink j position
    Move Sink j following it direction
    VNew=V with new sink position
    If VNew > VInit
      Permanently move sink to new position
    Else
      Come back to PositionInit
      Find new direction for the sink j
    End if
  End For
End Repeat
OFSP has finished
```

Figure 6: Pseudo code for OFSP

The sinks will tend to follow one specific direction. This is done to save some computation time in our simulation. To prove that our algorithm improved the WSN expectation life, some simulation results are presented. The WSN parameters are the following: Number of Sink: 5, Number of Node: 80 and field size: 9kmx9km. Figure 7 shows the consumption of the nodes as sinks are placed in different locations, determined by our optimization algorithm.

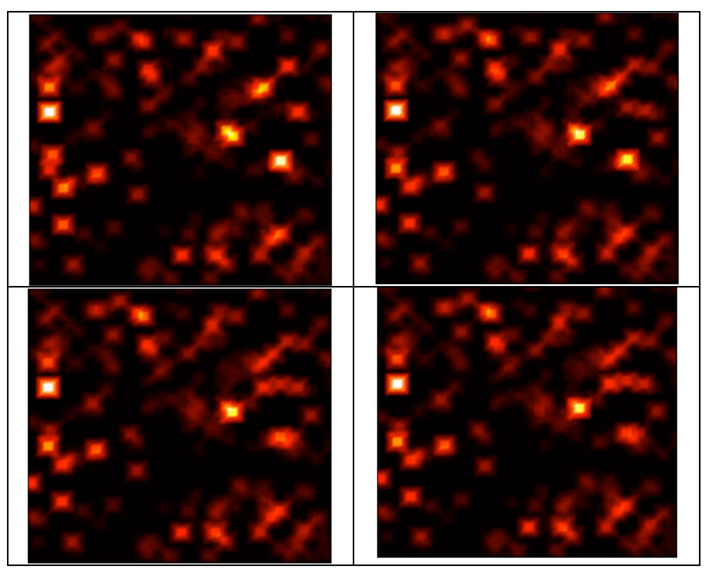

Figure 7: Evolution of the energy consumption of the nodes as sinks are placed in different locations.

As it has been explained on the section *life expectation of the WSN,* the overall consumption of the WSN does not determine the practical time that a WSN will run without outside intervention. This is due to the fact that some nodes may die a lot quicker than others. Therefore, the concept of Euclidian distance has been explained and is the parameter to optimize. The Figure 8 shows the improvement of V during the whole process. On the other hand, the overall consumption was constant during the algorithm (sometimes it may also increase a bit).

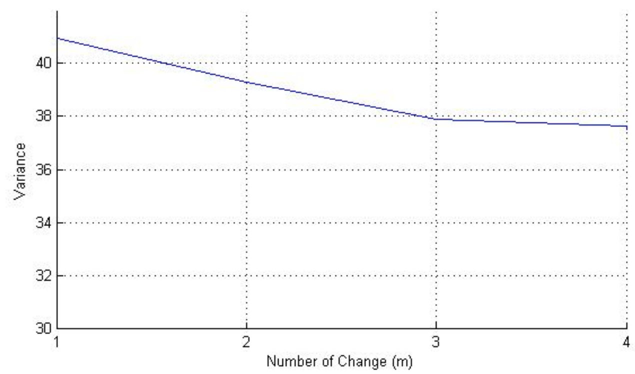

Figure 8: The OFSP V (Sum of the Euclidian distances between the consumption of each node and the average node consumption of the WSN) improvement for one simulation

6) Optimized nodes replacing sinks positioning

This situation is a bit trickier than when the sinks are placed everywhere because nodes are not able to follow a given direction. This is due the fact that the sinks cannot move anywhere but they have to replace existing nodes in existing location. Instead of the OFSP, we have decided to make use of a customized way to find better results than just in using the k-means method. After having chosen the initial sink location as described in 4), each initial sink will choose four neighbor nodes which are now potential candidates of being a sink instead of the initial position. This choice is made as a function of the distance between the nodes and the initial sink multiplied by a random factor as described below.

$$W_{bn} = D \cdot \left(X \cdot R_n + (1 - X) \right)$$

W_{bn}=weight for chosen the candidate neighbor
D=distance between sink and potential neighbor candidate
R_n=Random number between 0 and 1
X=number between 0 and 1 representing the importance of the randomness for W_{bn}.

```
Find the initial Sink Position with k-means

m=0

Repeat until none improvement of the variance

  m=m+1

  For i= 1 to NumberOfSink

    VarianceInit=Variance with actual sinks

    Find the X neighbours for sink(i)

    For y=1 to X

      VarianceNew=Compute the variance as
      neighbours(y) is the sink instead of sink(i)

      If VarianceNew>VarianceInit

        sink(i) = neighbours(y)

      End if

    End For

  End For

End Repeat

ONRSP has finished
```

Figure 9: the pseudo-code of ONRSP

The result of the algorithm improves almost every time the life expectation of the WSN. The example that we showed in this section have 4 sinks for 50 nodes. Figure 10 represents the evolution of the connection tree. The initial state is schematized by the upper right image. Then, the evolution of the tree is shown from left to right.

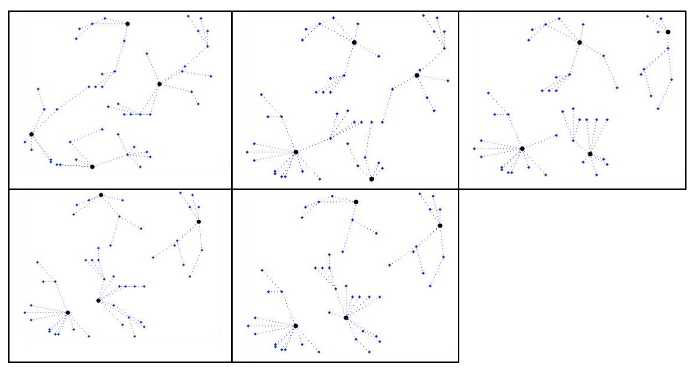

Figure 10: Evolution of the tree with ONRSP

The improvement of the WSN expected life is considerably higher than before (almost doubled). It is difficult to be more accurate because many parameters interact with it (as the multi-hop routing protocol, the exact type of data sent, and the sensor type). Using our algorithm, the overall consumption and V has improved on the example shown by Figure 10 as you may see in Figure 11 and Figure 12. It is interesting to note that the overall energy spent by the WSN may also increase slightly in some case when V is always decreasing (it is the goal of our algorithm).

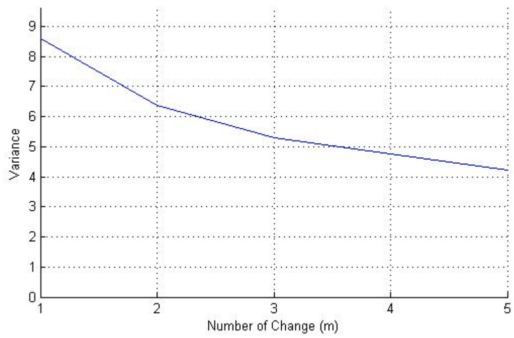

Figure 11: V improvement with ONRSP

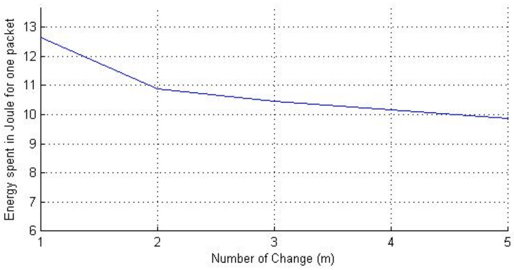

Figure 12: Overall energy spent improvement with ONRSP

V. RELATED WORK

There is an interesting discussion about Multiple-sink sensor networks in [10]. They developed a logical graph model to adapt a mono-sink to a multi-sink architecture, which decreases significantly the energy consumption of each node. However, they have not practically designed any protocols. In [15] we found a deep investigation about a system which describes an urban air pollution monitoring architecture with the sensors type GUSTO (Generic Ultraviolet Absorption Spectroscopy). In this paper, they make a difference between sensor networks and sensor grids. A sensor grid is the management of entire architecture and sensor network is about the physical connection between the sensors. Another interesting paper [16] describes a system for monitoring the air pollution with mobile sensors. The principal goal of this project was to establish maps of pollution variation in the city. Each sensor is coupled with a GPS device and a handheld computer. For the data collection, the device has to be synchronized to a desktop because no wireless module is implemented on the prototype. A multi-scale wavelet transforms [17] and self-organizing map neural network to mine air pollution data. The purpose of this paper is basically to find a good way to divide a sensor network into clusters. In [18], where an argument against data centralization is given, multi-hop routing is not used and the data are only accessible from a specific location. This type of architecture may be set up for military operation or disaster recovery for instance. Finally, [19] describes an architecture for archival storage. The particularity is that data are locally saved by each sensor because the energy cost of the wireless communication is a lot higher than the storage on a flash memory. Then, the sensor needs to send metadata which are indexed by the proxy.

About sinks optimal positioning: Only few articles have started investigations. [20] describes a method to select the most appropriate nodes to be sinks. However, they consider only the overall energy consumption of the WSN. Another interesting article [21] predicts the expected life time of a wireless sensor network and the optimum number of sink. They show the tree evolution over time.

VI. CONCLUSION

Our paper describes and discusses a platform collecting data generated by pollution sensors (ozone and nitrogen dioxine). The main contribution of our tools concerns the scalability of the proposed architecture. The size of usual sensor networks exceeds rarely 50 nodes and consequently do not need more than one sink. However due to the relatively low cost of the materiel, the scientific community is planning to deploy huge networks for monitoring critical data for the health care community. This kind of applications needs an adapted mechanism to collect and manage data. Our solution copes for the requirement of such large networks implementations. Another important innovation of our proposal concerns its adaptability to any sort of high level applications. The main difference with existing solutions is the presence of an embedded computer which shares the work and the intelligence with the centralized database server.

For sink positioning, we have described two algorithms to reduce the energy consumption, the Optimized Free Sinks Positioning (OFSP) and the Optimized Nodes Replacing Sinks Positioning (ONRSP). The contribution of these techniques is the fact that they optimize the sum of the Euclidian distances between the consumption of one node and the mean consumption of all nodes and not necessarily the overall consumption of the WSN. This way helps preventing the death of nodes which generally mean the disconnection of others nodes. The sinks positioning mechanisms, ONRSP and OFSP, improve significantly the WSN life expectation. However, only simulation has shown it. The next step would be to test it with a real WSN.

ACKNOWLEDGMENT

This work was supported by the MICS (Mobile Information & Communication Systems) and the IICT (Institute for information and communication technologies) at HEIG-Vd (Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud).

REFERENCES

[1] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks", In Proceedings of the 1st international Conference on Embedded Networked Sensor Systems (Los Angeles, California, USA,

November 05 - 07, 2003). SenSys '03. ACM Press, New York, NY, 14-27.
[2] J. Hill and D. Culler, “Mica: A wireless platform for deeply embedded networks”, IEEE Micro, vol. 22, no. 6, pp. 12–24, 2002.
[3] J. Ibriq and I. Mahgoud, "Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges", Proceedings of the 2004 Symposium on Performance Evaluation of Computer Telecommunication Systems
[4] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks", IEEE INFOCOM 2003 - The Conference on Computer Communications, vol. 22, no. 1, Mar 2003 pp. 1713-1723
[5] Q. Li, J. Aslam and D. Rus, "Hierarchical Power-aware Routing in Sensor Networks", In Proceedings of the DIMACS Workshop on Pervasive Networking, May, 2001
[6] I. Solis and K. Obraczka, "The Impact of Timing in Data Aggregation for Sensor Networks", In Proc. of the IEEE International Conference on Communications (ICC), 2004.
[7] R. Cristescu, B. Beferull-Lonzano and M. Vetterli, "On Network Correlated Data Gathering", In Proc. of the 23rd Conference of the IEEE Comunications Society (INFOCOM), 2004.
[8] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks", in International Workshop of Distributed Event Based Systems (DEBS), Vienna, Austria, July 2002.
[9] H. Dubois-Ferriere and D. Estrin, "Efficient and Practical Query Scoping in Sensor Networks", Tech. Rep. 2004--39, CENS/UCLA Tech Report, 2004.
[10] A. Das and D. Dutta, "Data acquisition in multiple-sink sensor networks", SIGMOBILE Mob. Comput. Commun. Rev. 9, 3 (Jul. 2005), 82-85.
[11] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks", In Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing (Lausanne, Switzerland, June 09 - 11, 2002), MobiHoc '02, ACM Press, New York, NY, 194-205.
[12] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network". In Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking (Seattle, Washington, United States, August 15 - 19, 1999). MobiCom '99. ACM Press, New York, NY, 151-162.
[13] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks", In IEEE Wireless Comm. and Networking Conference (WCNC 2003), 2003.
[14] H. Dubois-Ferrière, R. Meier, L. Fabre, P. Metrailler, "TinyNode: A comprehensive platform for WSN applications", LCAV-REPORT-2005-003
[15] M. Ghanem, Y. Guo, J. Hassard, M. Osmond and R. Richards, "Sensor Grids for Air Pollution Monitoring", UK e-Science All Hands Meeting, Nottingham UK, September 2004
[16] A. Steed, S. Spinello, B. Croxford and C. Greenhalgh, "e-Science in the Streets: Urban Pollution Monitoring", UK e-Science All Hands Meeting, September 2003
[17] S.-T. Li and S.-W. Chou, "Multi-resolution spatio-temporal data mining for the study of air pollutant regionalization", In Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.
[18] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A.L. Murphy, and G.P. Picco, "TinyLime: Bridging Mobile and Sensor Networks through Middleware", In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom), Kauai Island, Hawaii, USA, March, IEEE Computer Society, pp. 61-72, 2005
[19] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: a two tier sensor storage architecture using interval skip graphs", In Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems (San Diego, California, USA, November 02 - 04, 2005). SenSys '05. ACM Press, New York, NY, 39-50.
[20] H. Kim, Y. Seok, N. Choi, Y. Choi, and T. Kwon, "Optimal Multi-sink Positioning and Energy-efficient Routing in Wireless Sensor Networks", Lecture Notes in Computer Science (LNCS), Vol.3391, pp.264-274, Springer-Verlag, January/February 2005
[21] E. I. Oyman and C. Ersoy, “Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks”, Proceedings of the International Conference on Communications (ICC 2004), Paris, France, June 20-24, 2004.
[22] M. Hata, "Empirical formula for propagation loss in land mobile radio services", IEEE Trans. Veh. Technol., vol. VT-29, pp. 317--325, Mar. 1980.
[23] R. Min and A. Chandrakasan, "A framework for energy-scalable communication in high-density wireless networks", In Proceedings of the 2002 international Symposium on Low Power Electronics and Design (Monterey, California, USA, August 12 - 14, 2002). ISLPED '02. ACM Press, New York, NY, 36-41
[24] E. W. Dijkstra, "A note on two problems in connexion with graphs", In: Numerische Mathematik. 1 (1959), S. 269–271
[25] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, 1967