

A new management method for wireless sensor networks

A. Jacquot, Jean-Pierre Chanet, K.M. Hou, Gil de Sousa, A. Monier

▶ **To cite this version:**

A. Jacquot, Jean-Pierre Chanet, K.M. Hou, Gil de Sousa, A. Monier. A new management method for wireless sensor networks. 9th IEEE IFIP Annual Mediterranean Ad Hoc Networking Worshop, Jun 2010, Juan les Pins, France. 9 p. hal-00520485

HAL Id: hal-00520485
https://hal.science/hal-00520485

Submitted on 23 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author-produced version of the paper presented at th IEEE IFIP Annual Mediterranean Ad Hoc Networking Worshop
June 23-25,2010, Juan les pins, France. Original paper avialable at http://ieeexplore.ieee.org/

A New Management Method for Wireless Sensor Networks

Aurélien JACQUOT*, Jean-Pierre CHANET*, Kun Mean HOU§, Gil De Sousa* and Antoine Monier§
*Cemagref, UR TSCF, 24 avenue des Landais 63170 Aubière, France
Email: {aurelien.jacquot, gil.de-sousa, jean-pierre.chanet}@cemagref.fr
§ISIMA - LIMOS UMR 6158 CNRS, Université Blaise Pascal 63170 Aubière, France
Email: {kun-mean.hou}@isima.fr, antoine.monier@anelis.isima.fr

***Abstract*—The Wireless Sensor Networks (WSN) with their constant evolution, need new management methods to be monitored efficiently by taking into account the context and their constraints such as energy consumption, reliability and remote monitoring. WSN has diverse application domains: smart home, smart care, environmental data collection etc. In order to manage a large scale WSN, several Wireless sensor network Management Tools (WMTs) are developed. Some of them use SNMP protocol like because it is impossible to implement the full compliance classical SNMP standard on each wireless sensor node. Therefore, it is important to develop a new WMT with a restricted SNMP standard dedicated to WSN applications. In this paper, we present a new WMT named LiveNCM: LiveNode Non invasive Context-aware, and modular Management tool. LiveNCM is divided into two main parts: one is centralized on the fixed network structure and another one, distributed on each node. Each part introduces the concept of non-invasive context-aware to reduce data exchanges and diagnoses the wireless sensor node state with few messages. Moreover, nodes are based on a configurable modular architecture enables to adapt to an application and to a local node constraints. LiveNCM is implemented on the LiveNode platform to validate the energy consumption and on a UNIX system to validate the adopted SNMP sub-agent. Ultimately, a decreasing data exchange and an improvement in the energy consumption in the entire WSN were observed. An implementation of the proposed management method is presented. *Index Terms*—ireless Sensor Network, ZigBee, Network Management, Environmental data collection-less Sensor Network, Zigbee Network Management, Environnemental data collectionireless Sensor Network, ZigBee, Network Management, Environmental data collection-less Sensor Network, Zigbee Network Management, Environnemental data collectionW**

I. Introduction

The various technological advances in recent years drive a new research domain in embedded systems: Wireless Sensor Networks (WSN) [1]. A WSN is a set of wireless sensor nodes where each node senses physical values and sends them to a supervisor or users via specific nodes named sink nodes. The potential of such network is huge and several kinds of applications could be design [2] [3] [4]: military [5], industrial [6], transport [7], agriculture [8] [9], environmental survey [10]... The variety of applications reveals many different constraints that reduce real world application deployments. A correct use of a wireless sensor node induces an intelligent management of its three main resources such as energy, memory and processing. Moreover, the WSN application constraints can be classified into different categories:

1) *Communication*: wireless access medium range and bandwidth (QoS: Quality of Service);
2) *Environment*: the application is tightly coupled with the indoor or outdoor environment context (e.g. hostile: corroded, radioactive, polluted);
3) *Measurement*: the measurement includes all constraints related to the sensor like the calibration, the data quality or the parameters of normal functioning.

Besides, each constraint is directly or indirectly related to one or more wireless sensor resources. For example, the communication protocol must be chosen to take into account the available energy on the sensor because communication represents 75 % of the energy cost of nodes.

Without an efficient Wireless Management Tool (WMT) [11], it is difficult to use this kind of technology. Therefore, since few years, the main objective of researches on WSNs is to define a practical and user-friendly management tool [12].

The first idea is to translate a wire management protocol like SNMP standard [13] in a wireless management tool for WSN. ANMP [14], GUERILLA [15] and SHAMAN [16] implement and use SNMP on nodes to take management information. The main disadvantage of these methods concerns the memory size on each node and the CPU features. Indeed, the SNMP standard was designed for monitoring wire networks having TCP/IP protocol stack, thus it has many sub-section dedicated to communication protocols (e.g TCP or ICMP protocols). Moreover, another work named SNMP Proxy [17] uses an SNMP agent to propose a common users' interface. The architecture was based on the CodeBlue application [18], which centralize data and makes a log file. A new software translates those files into give information to the SNMP agent. The main disadvantage of this method is the unidirectional communication, from WSN to the agent, and they introduce a new delay between collected data and shown to users. Unlike a wire network, WSN needs more information on the context and the quality of link between two nodes. The advantage of these methods is to use an easily and known standard with many visualization tools.

The second kind of tools is not based on SNMP. We can distinguish two categories of WMTs: i) Middleware category like IMPALA [19] and MATE [20]; ii) Specific method category with SNMS [21] or Node energy level manager method [22]. The first category was based on a middleware to provide

a modular interface for software deployment. On one hand, their main advantage is to enable an easily adaptation of an application to the local constraints and processing. On the other hand, they need more memory and processor resources. The second category was design for specific treatments like energy. The main disadvantage is the lack of modularity and of evolution of these methods, it is necessary to redesign a new method to take into account a new constraint such as energy consumption. These categories were design for a specific treatment or to facilitate the software deployment. However, there are not real management tools to answer to the large scale WSN problematic.

Thus, our work is based on SNMP protocol and we propose a WSN extension taking into account the WSN constraints. The remainder of this paper is organized as follows. In the next section, our new wireless management tool will be described. Section 3 presents materials and methods to evaluate our work, and Section 4 presents the obtained results and the discussion about the combine of SNMP protocol in WSNs and estimators. Finally, we conclude the paper with an assessment of our implementation and present the ongoing work.

II. A New Management Method

Created in 1990, SNMP standard [13] is very popular in many wired network to manage computer peripherals and, many tools exist to manage information and to configure remotely a network. Therefore, it seems interesting to use SNMP like method to have a user-friendly application to manage a WSN, mainly due to the wireless sensor resources limitations. Unfortunately, SNMP cannot be used as it is because it is not adapted to WSN constraints.

We propose a new technique, named LiveNCM stands for *"LiveNode Non invasive, Context-aware and modular Management tool"*. This technique is developed to meet network management and saving energy demands in WSN. The propose solution has two main components:

- the first one regroups a fixed and dedicated or shared server with an extended SNMP agent using an estimator model and coupled to a gateway to communicate with the WSN (see Fig. 1);
- the second one is embedded on the wireless nodes, it is based on the same methods but adds a modular programming concept to meet the node constraints.

The following section details the different functions of our management method.

A. SNMP Sub Agent

SNMP implements an agent that performs and stores management data, and responses to user requests. To extend an agent, it is possible to rewrite a new agent, to create an autonomous agent, or to extend the SNMP agent by a plug-in named proxy. In LiveNCM, the second solution was adopted to unload the SNMP agent already running on the server. This method permits to increase the fault tolerance on the server because the LiveNCM sub-agent limits its influence on the current SNMP agent. This sub-agent is based on its data

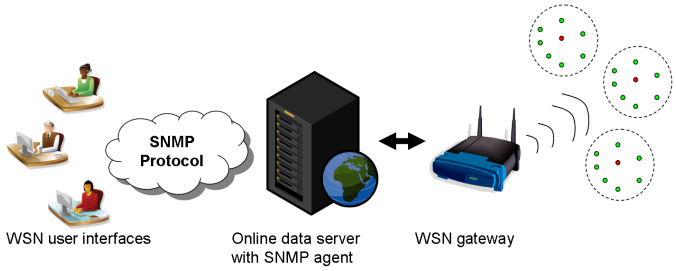


Fig. 1. Overview of the LiveNCM management method

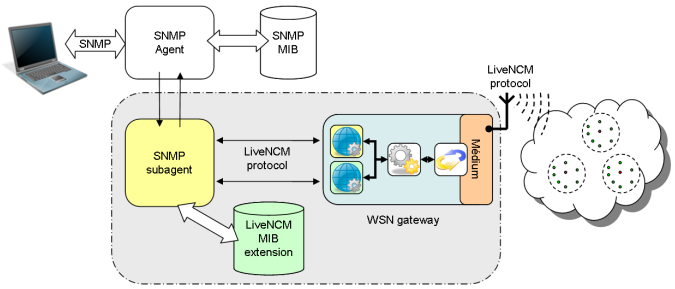


Fig. 2. LiveNCM deployment architecture

structure (MIB) that will detail in the next sub-section, and is fully autonomous on the server. When the first agent received a user request and it cannot respond, the request is treated by the appropriate sub-agent. LiveNCM sub-agent interprets requests and responds when it is possible, otherwise, it forms a WSN request and sends it through TCP or UDP port to the WSN Gateway (described in a next sub-section). The Fig. 2 illustrates the architecture of the LiveNCM tool.

B. SNMP MIB Extension

A SNMP agent performs and stores management data in a Management Information Base (MIB). The most used MIB is defined by the RFC 1213 [23]. Four major generic sub-trees are defined: "*mgmt*" sub-tree has all basic and common information; "*private*" sub-tree contains all private sub-trees of manufacturing devices; "*experimental*" and "*directory*" sub-trees regroup all others data. So, the extension of the MIB for the LiveNCM sub-agent on the server takes place under the "*private*" sub-tree. Each WSN is represented by two sub-trees: the first branch stores information about WSN description. The second one regroups four categories of management data. Each category is represented in the MIB by a table with one row by nodes. The Fig. 2 shows the LiveNCM MIB structure.

The "*nodesDescTable*" table regroups information about names, IDs and location of nodes, whereas the "*nodesCfgTable*" table stores the running mode, the role of a node in a WSN, and its current state. Then, in the "*nodesComTable*" table, the sub-agent contains the node transmission configuration of nodes such as IP, transmission power, transmission channel and some network statistics (send/receive traffic, errors rates...). The last table "*nodesSensorTable*" is used to

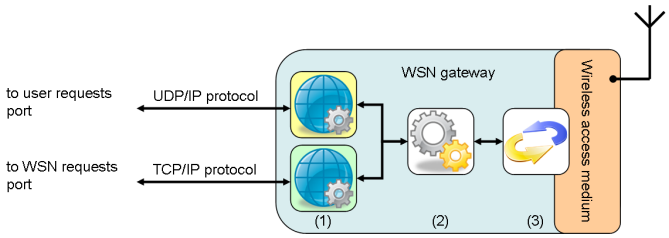


Fig. 3. LiveNCM gateway design

manage sensors parameters such as the last sample value, the sample rate and information about estimators. To reduce data exchange, the indirect diagnosis principle based on estimator was integrated in the sub-agent for this table. Users can, for instance, supervise the battery level at any rate because the sub-agent estimates a value between two received values. Users may force to request a real sample value, but sometimes, the WSN is in a sleep mode and the request must be scheduled to the next wake up of nodes. Thus, it is difficult to get a real sample value at any time and respect the energy constraints.

C. WSN Gateway

The LiveNCM sub-agent transforms SNMP user requests into WSN requests. To transfer these requests, LiveNCM introduces a gateway deployed on the server or on a computer near the wireless sensor network. Its main role is to transfer requests between the sub-agent and the WSN. The Fig. 3 presents the architecture of this gateway. A SNMP request is translated in a WSN requests, sent to this gateway by a TCP or UDP port, and directly transferred to the node. In the other way, messages of nodes may be processed by the gateway to make some mathematical operations like the average or median, and/or be transferred to the sub-agent. The sub-agent will process the received message to fill the MIB structure and responds to user request.

A WSN may be deployed on a large place, and it is not possible to communicate with all nodes from the server executing the gateway. Therefore, we can distinguish three ways, presented on the Fig 4, to deploy the gateway in an existing network. The first one (A) is to use a remote computer with the gateway software and a communication interface, using in our example the ZigBee protocol, to send requests to the WSN. Each part, SNMP server, WSN gateway, and user applications can be connected on a WAN or LAN. The second way (B) is to use the same computer to run a SNMP server and the WSN gateway. These two solutions to cover the WSN network, need power supply, wired network and a building nearby to host the additional computer. Unlike these deployments, the third architecture (C) can be based on a WiFi/ZigBee proxy to implement the WSN gateway. The proxy is integrated in the network, and users and SNMP server can exchange information on a LAN or WAN network. This solution is autonomous, so the proxy can be deployed in the WSN network. In the worst case, where there is no wireless network within range, a GSM modem interface can replace

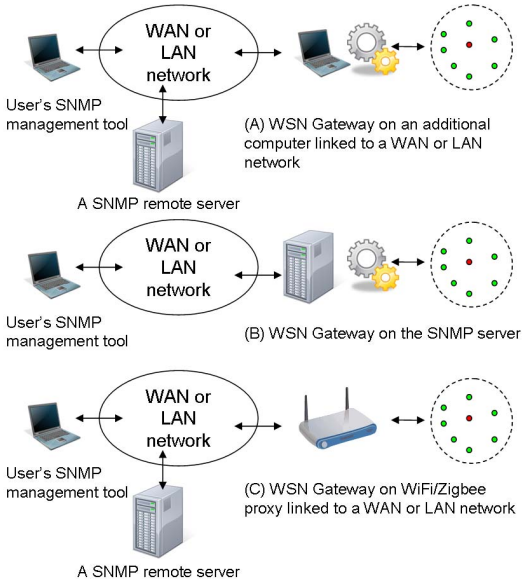


Fig. 4. The different way to deploy the WSN gateway

WiFi device. Thus, the proxy with GSM is closed to the (B) deployment (the gateway program is running on the SNMP server and uses the GSM interface).

Obviously, to make it possible, we have chosen to develop and share the same code for nodes and gateway. So, to deploy the third solution on a WiFi/ZigBee node, user can use the software of the gateway and quickly adapt it to the node's constraints (energy, communication medium).

D. Indirect Diagnostic and Estimator

The most important challenge for a WSN management method is to preserve the energy in the nodes to increase their lifetime. In wireless sensor, the transmission medium is the main cause of energy consumption, so it is necessary to minimize message sending on each node. We can distinguish different methods to reduce message sending in a WSN. The first technique frequently applied in many management tools is data compression. The information is compressed and sent at the same rate than uncompressed information. A second method consists to make an indirect diagnosis of the current node state with few messages. LiveNCM implements this second approach.

In many situations by taking into account the context, with a received message, a system can diagnose the sender state. The concept named *Indirect Diagnosis* is implemented in LiveNCM, and with only one message, a receiver can detect a time fault, and estimate battery level. Moreover, with measurement knowledge, it may be able to detect the quality and the validity of the measurement chain. Thereby, with only one message, we are able to extract some information, concerning the state of the node. Indirect diagnosis uses all common exchanged messages to extract information, and reduces management message requirements. If we consider

three additional data by message, the data management exchange may use few additional messages and uses three times less messages than without indirect diagnosis to take the same management information. As a result, without real compression, data exchange in a WSN is reduced, and the node lifetime may increase.

Another way to reduce the quantity of data exchange is to use models to predict data evolution [24]. Environmental data have a relatively slow evolution (soil moisture, temperature...) and they are well adapted to this kind of approach. So, LiveNCM integrates an estimator model, which enables to send only the pertinent sample data. Estimator can use different models, like quantization model or 2 or 3-order polynomial model for example. The quantization model is the most simple but its precision is very bad, because it uses few samples and while errors are under a fixed threshold, the estimated value is fixed. Models which use 2 or 3-order polynomial are more precise but they use many processor and memory resources to compute a new model and more sample data are needed to be sent. Nevertheless, a first order model (linear model) is efficient to estimate a slow signal evolution and uses few processor and memory resources. Thus, LiveNCM implements a first order polynomial model. It is based on the last sending value (t_S, V_S) and the last sample value (t_R, V_R), and an estimator is compute with these values using the following formula: $G = \{(t_R, V_R), (t_S, V_S) | V_R = a * t_R + b$ and $V_S = a * t_S + b\}$ where the a and b coefficients are computed when is necessary to send the sample data. The sending threshold is fixed by user or by an application to have a compromise between precision and the amount of sending data. When the difference between the sample value and the estimated value is greater than this threshold, the node sends the sample data and updates the coefficients of the estimator model. Therefore, it implies two levels of knowledge in a WSN; the slave node has the sample data and the estimation model, whereas the master node or the supervisor needs only the estimated sample data.

Otherwise, the LiveNCM estimator contains a periodic synchronization to minimize estimated errors. This method sends additional sample data to synchronize and update the estimator model. In some case, with a low frequency signal, the estimator may give an alert on a threshold but on the node, everything is all right. For example, a battery discharge evolves very slowly and without synchronization, errors are too high and user software can detect an energy fault while the battery level is medium. The second interest of this synchronization is to have a periodic map of actives nodes. Without any message, a supervisor can detect a fault on a node when it received no periodic message. Moreover, on a large WSN, it is possible to schedule these periodic messages in order to reduce traffic collisions and network charge.

III. Materials and Methods

A. The LiveNode Platform

The nodes used are based on a LiveNode platform presented on the Fig. 5 [25]. The node has more CPU and memory than



Fig. 5. LiveNCM gateway design

Processor	ATMEL SAM7S256 (ARM7TDMI)
Frequency	500 Hz to 48 MHz
Flash memory	256 Kb
Ram Memory	64 Kb
Access medium	ZigBee IEEE 802.15.4
Transmission range	Indoor: up to 100 m Outdoor (LOS): up to 1.5km

TABLE I
LiveNode features

MICA motes [26] or TinyNode [27]. The table I presents the LiveNode's features, which are detailed in [28]. Embedded memory can be used to implement more complex application and to perform pre-processing treatments. In order to reduce energy consumption, the LiveNode platform has a real time clock with a sleep and wake-up system. Most of the time, a node is in SHUT-DOWN mode to save energy, and when it needs to take a measure, it is in the MEASURE mode. Then, for few seconds, the node turns on the ZigBee medium to send data or to receive requests from the manager node. All energy consumption modes are described in the table II. As a result, it can be noticed that the TRANSMISSION energy mode must be minimized to preserve the node lifetime.

B. The Agent LiveNCM

The LiveNCM agent was developed in C with the Net-SMNP Library [29] that proposes a set of tools to make a new SNMP agent. So, we realized a MIB extension with this associated sub-agent. It is implemented by 3 modules:

- a *structure module* to keep the link between the real network structure and the data structure of the agent. To delete a node for example, the sub-agent must delete the associated sensors to maintain the MIB structure in accordance with the network structure. So, a generic data structure is used as a container for the SNMP data

Functioning mode	Energy consumption
IDLE	40 mA
MEASURE	50 mA
TRANSMISSION	120 mA
SHUT-DOWN	0.2 mA

TABLE II
Energy modes of LiveNode platform

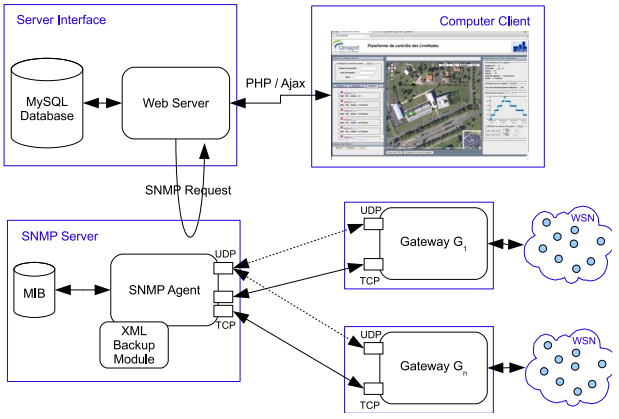


Fig. 6. Supervisor Architecture

structure to quickly access to information and elements of a node or a WSN;

- a *communication module* to manage users SNMP requests and to process messages from network elements. This module processes messages and requests, from users to network elements and from network elements to the sub-agent, and from users to the sub-agent;
- a *backup module* to manage the restart of the agent. Normally, a SNMP agent is designed to store a snapshot of a system in its process memory. When an agent restarts after a crash, it can inspect easily and quickly the status of the equipment because it is embedded on it. In LiveNCM, the agent must talk to the networks to obtain all the information. So, in case of communication unavailability, we choose to implement this module to store the MIB in XML files at periodic times to keep a memory of the network state in case of problem with the agent.

A supervisor system has been developed to manage the network and to visualize the sensor data. The system uses the agent to connect to the WSN via gateways. A web interface allows the use of the agent in a friendly way. The interface is made up of an Apache web server, and a MySQL Database to store the sensor historic. Moreover, we develop a script to maintain the MySQL database with the SNMP MIB information. The interface is a geographic one based on Google Maps. The Fig. 6 presents the different modules of the supervisor architecture. The interface with the database is separated of the sub-agent to allow everyone to develop its own interface with their functionalities. In this case, the SNMP agent, the database and the gateway can be deploy on different sites.

C. The Evaluation Context

To evaluate the SNMP sub-agent and the estimators, we configure the LiveNode platform to perform sample data with a frequency of 100Hz. The ATMEL Analog to Digital Converter (ADC) of LiveNode provides two precisions: 8 and 10 bit. To reduce the noise due to the measurement chain, the ADC peripheral performs 10-bit resolution data that we convert into 8 bit data. At each sample (period of 10 ms), the processor verifies the quality of the estimated value, and when errors are too high, the node computes new estimator coefficients and sends the last sample value to his WSN gateway. A signal sinusoidal or triangular signal at different frequencies is connected to an ADC channel of the processor and we fixed the estimator threshold to 0.16 V (4 times more than the ADC resolution). The main goal of this experimentation is to find the input admissible frequency signal that can be produced by the estimator. The second part of LiveNCM is running under Linux Ubuntu 8.10 with its current SNMP daemon. A LiveNCM sub-agent was developed with the NET-SNMP framework and it uses an UDP port to transfer requests to the WSN gateway. This gateway is written in C code, and transfer requests of sub-agent to a node. The language was chosen to be shared and to facilitate the code development between the LiveNode and the Linux OS platforms. In the other way, the gateway treats WSN responses to make SNMP queries. The SNMP agent with the LiveNCM sub-agent, and the WSN gateway were embedded on the same computer.

IV. Results and Discussion

A. SNMP Data Exchange

SNMP was designed for wire network; as a result, it does not take into account the resource constraints (memory and CPU etc.). In case of a WSN, the Protocol Data Unit (PDU) must be reduced to minimize the network overload. Thus, LiveNCM chooses to use its own protocol between the gateway and WSN. For example, a typical SNMP SET-REQUEST needs 50 to 100 bytes. Moreover, the SNMP SET-REQUEST is completed by the GET-RESPONSE request, which needs the same amount of bytes. Finally, to send a management data to a node with SNMP standard, 100 to 200 bytes of the exchange data is necessary. Thus for a WSN of 500 nodes, the amount of the exchange data size may reach 50 to 100 KB. Unlike the SNMP standard, a LiveNCM set-request needs only 13 to 17 bytes and its response uses around 25 bytes for one sensor value. Thus for a WSN having 500 nodes, the exchange data size is around 19 to 21 KB instead of 50 to 100 KB. Otherwise, for large network, we can use the IPv6 protocol to address nodes and SNMP agents. Indeed, the IPv6 protocol can address 4 times more devices than the traditional IPv4 protocol. For WSN applications, we try to reduce data exchanges and it is interesting to use and make compatible our SNMP agent with the IPv6 6LowPAN protocol [30]. This protocol compresses IPv6 PDUs header into few bytes. Therefore, we adopt some of the key 6LoWPAN features in the WSN protocol used in LiveNCM. For example, with one byte, we can pass the maximum hop-numbers, the type of PDU, and the priority. Each bit in a byte can contain information about the node or the sent PDU. The main idea is to use all sending bits including an information about the state of the node or of the network.

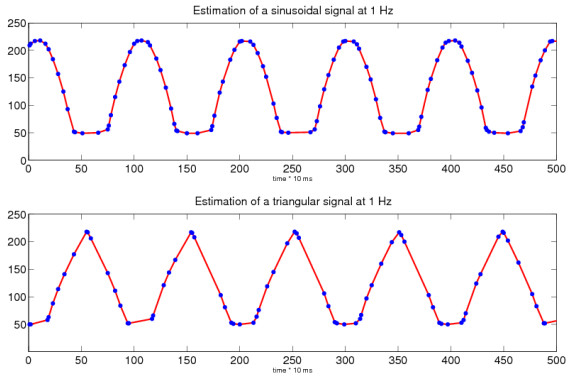


Fig. 7. Interesting samples of a sinusoidal (top) and a triangular (bottom) signal at 1 Hz with a threshold of 0.16 V

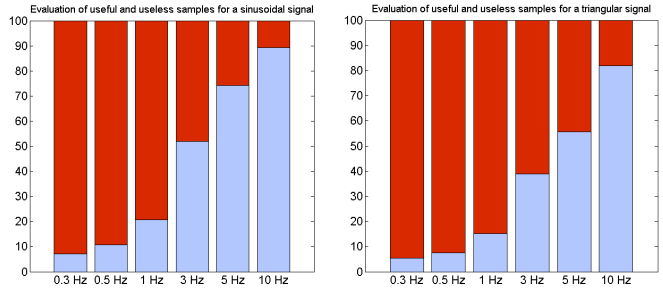


Fig. 8. Comparison between samples values taken (black) and sent (grey) for a sinusoidal (left graph) and a triangular (right graph) signal

B. Signal Acquisition and Estimation

The first evaluated signal is a sinusoidal signal at different frequencies. On the Fig. 7, we represent the estimated signal of a 1Hz sinusoidal signal, which uses 20% of sample data. The sample frequency was set to 100Hz, which represents 100 samples per second. With the 1Hz signal, LiveNCM divides by 6 the data exchange and uses only 15 sample data per second. For other frequencies, presented on the Fig. 8, a 0.3Hz signal divides the traffic by 20 and uses only 5% of sample data (Fig. 8). Note that at higher frequency (10Hz), the LiveNCM estimator uses 93% of samples but estimation errors are higher than a slower signal. Moreover, in a WSN, it is usually to assign 75% of energy consumption to transmission. So, with LiveNCM, the autonomy of a node may be multiply by 3 for the lowest frequency signal (e.g. temperature). In a second time, we evaluate the estimator on a triangular signal having the same frequencies as the sinusoidal signal. This type of signal is adapted to our estimator because there is only linear parts, thus, the performances may be better. An estimation of a 1 Hz triangular signal is presented on the Fig. 7 and the Fig. 8 represents the comparison between samples taken and sent. Therefore, it can be noticed that the useful rate is lower than the first signal, and the autonomy of node are better. For example at 1Hz, it needs only 15% instead of 20% for the first signal.

Finally, the LiveNCM estimator based on a linear model decreases the data exchanges on a WSN. For example, on a network of 100 nodes, where each node takes a measure of the slowest sinusoidal signal at 100Hz, and each message represents 10 bytes, the data traffic can reach a rate around 100KB/s. With LiveNCM, each node sends only 5% of samples and the data traffic can reduce to around 5KB/s. Furthermore, the environmental data such as temperature, pressure, or soil moisture, evolve smoothly, and then it increases the performances of the embedded estimator because there is more linear part in the signal.

C. Deployment on an Environmental Network

To evaluate the LiveNCM management tool, we deploy it on an environmental network to monitor information about a farming field such as the temperature and soil moisture. This network is composed of LiveNode sensors with a ZigBee communication module and data exchanges are made by one hop message. So, the sink node can access to information of any node in the network at any time. Some nodes are equipped with temperature sensor and/or with a GPS module for mobiles nodes. The transmission frequency is set by the web interface to 10s for all GPS nodes and 1min for others. The Fig.9 represents samples data collected by a node with an embedded sensor. The top graph compares estimated and real values and the second graph presents all sample values sent. Circle points are samples values that must be sent and star points are synchronization sample values. With a threshold fixed to 0.5C and a synchronization period of 1 hour, we note an error around 2.5% and the node sends only 7% of all samples values (150 values sent for 2000 values taken). Moreover, each message has 13 bytes to form its header/footer and uses 9 bytes to send the sample value and the battery level. So, on experiments, nodes with a temperature sensor send around 44KBytes of samples values. With the estimator activated, this volume decreases to 3.3KBytes. In this case, we decrease collision and sink node blocking; so, it is possible to multiply the number of active nodes managed by the same sink node.

V. Conclusion

In this paper, we present a new management technique for WSN based on SNMP like tool: LiveNCM integrating a simple estimator model, and our communication protocol, which enable to minimize message exchanges. Thereby it enables to increase the WSN lifetime. Note that LiveNCM is compatible with standard SNMP supervisor that permits a large panel of monitoring application. Moreover, the diagnosis indirect method integrated in this WMT reduces the exchange data volume in a large scale WSN. On one hand, the obtained results show that with only 5% of sample data in case of a

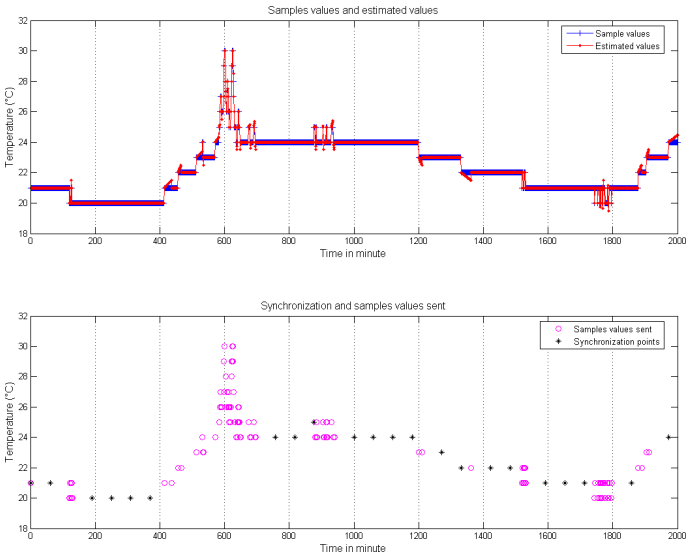


Fig. 9. Temperature monitoring with a LiveNode implementing a 1-order polynomial estimator

sinusoidal signal (frequency of 0.3Hz), the estimated signal remains good with few errors. With others types of low frequency signals, like battery discharge, temperature or soil moisture, our estimator reduces also the data exchanges in the WSN. For example, on a battery discharge, we divide by 4 the number of data samples. On the other hand, without embedded SNMP agent on nodes, the management needs on nodes small memory footprint, and deports management data storage on a server without resource constraint. Communications, between the gateway and nodes, are made with a simple protocol defined on some features of standard ones. A real implementation on a WSN dedicated to environmental survey has been presented and validated.

The next step in our work will be the deployment on an application for a large period to evaluate more precisely the energy gain and the efficiency of LiveNCM. We will also focus our work more on an integration of a compression algorithm like the Huffman algorithm or the algorithm used in 6LowPan [30] for IPv6 low power network. Some features of this protocol seem interesting to be implemented in the LiveNCM protocol, to reduce more and more data exchanges. We have also to integrate a routing protocol, like CIVIC [31], on the wireless sensor nodes in order to evaluate the global performance of the networks.

Acknowledgement: The authors would like to thank the "Conseil Régional d'Auvergne" for its financial support to this project.

REFERENCES

[1] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proceedings of the IEEE*, vol. 91, 2003, pp. 1247–1256.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," in *Comput. Networks*, vol. 38, 2002, pp. 393–422.

[3] D. Puccinelli and M. Haenggi, "Wireless sensor networks: Applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, Salt Lake City, UT, USA, 2001, pp. 2033–2036.

[5] S. S. Iyengar, A. Tandon, and R. R. Brooks, *An overview*, ser. Chapman and Hall - CRC Computer and Information Sciences. S. S. Iyengar and R. R. Brooks, 2005.

[6] X. Shen, Z. Wang, and Y. Sun, "Wireless sensor networks for industrial applications," in *Fifth World Congress on Intelligent Control and Automation*, vol. 4, Hangzhou, China, 2004, pp. 3636–3640.

[7] H. Zhou, G. Desousa, J. Chanet, K. Hou, J. Li, C. Devaulx, and M. Kara, "An intelligent wireless bus-station system dedicated to disabled, wheelchair and blind passengers," in *IET International Conference on Wireless, Mobile and Multimedia Networks ICWMMN 2006*, Hangzhou, China, 2006, p. 4.

[8] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 50–57, 2007.

[9] N. Wang, N. Zhang, and M. Wang, "Wireless sensors in agriculture and food industry - recent development and future perspective," *Computers and electronics in agriculture*, vol. 50, no. 1, 2006.

[10] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," pp. 108–120, 2005.

[11] Z. Li, X. Zhou, S. Li, G. Liu, and K. Du, "Issues of wireless sensor network management," *Lecture Notes in Computer Science*, vol. 3605, pp. 355–361, 2005.

[12] W. L. Lee, A. Datta, and R. Cardell-Oliver, "Network management in wireless sensor networks," in *Handbook of Mobile Ad Hoc and Pervasive Communications*, D. M. K. and Y. L. T., Eds. USA: American Scientific Publishers, 2006, pp. 1–20.

[13] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol," vol. RFC 1157, 1990, pp. 1–36.

[14] W. Chen, N. Jain, and S. Singh, "Anmp: Ad hoc network management protocol," in *Selected Areas in Communications, IEEE Journal on*, 1999, p. 26.

[15] C. Shen, C. Jaikaeo, C. Srisathapornphat, and H. Huang, "The guerrilla management architecture for ad hoc networks," in *MILCOM 2002*, vol. 1, 2002, pp. 467–472.

[16] A. Sethi, D. Zhu, and P. Kalyanasundaram, "Shaman - an environment for distributed management applications," in *Integrated Network Management Proceedings*, Seattle, USA, 2001, pp. 321–324.

[17] L. Yen Yang, M. Messina, F. Kargl, L. Ganguli, M. Fischer, and T. Tsang, "Snmp proxy for wireless sensor network," in *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, 2008, pp. 738–743.

[18] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "Codeblue: An ad hoc sensor network infrastructure for emergency medical care," in *International Workshop on Wearable and Implantable Body Sensor Networks*, 2004, pp. 1–4.

[19] T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," in *Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming*. San Diego, California, USA: ACM, 2003, pp. 107–118.

[20] P. Levis and D. Culler, "Mate: a tiny virtual machine for sensor networks," in *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002, pp. 85–95.

[21] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, 2005, pp. 121 – 132.

[22] G. Jolly, M. C. Kus, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," *Computers and Communications, IEEE Symposium on*, vol. 0, p. 335, 2003.

[23] K. McCloghrie and M. Rose, "Management information base for network management of tcp/ip-based internets: Mib-ii," vol. RFC 1213, 1991, pp. 1–70.

[24] C. Q. Zhang, M. L. Li, and M. Y. Wu, "A model-aided data gathering approach for wireless sensor networks," *Journal of Information Science and Engineering*, vol. 23, no. 4, 2007.

[25] K. M. Hou, G. De Sousa, J. P. Chanet, H. Y. Zhou, M. Kara, A. Amamra, X. Diao, C. De Vaulx, J. J. Li, and A. Jacquot, "Livenode: Limos versatile embedded wireless sensor node," *Journal of Harbin Institute of Technology*, vol. 39, 2007.

[26] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," in *Technical report*, U.C. Berkeley, 2001, pp. 1–12.

[27] H. Dubois-Ferriere, R. Meier, L. Fabre, and P. Metrailler, "Tinynode: A comprehensive platform for wireless sensor network applications," in *Proceedings of the fifth international conference on Information processing in sensor networks IPSN 2006*. Nashville, Tennessee, USA: ACM Press, 2006, pp. 358–365.

[28] K. M. Hou, G. De Sousa, J. P. Chanet, H. Y. Zhou, M. Kara, A. Amamra, X. Diao, C. De Vaulx, J. J. Li, and A. Jacquot, "Livenode: Limos versatile embedded wireless sensor node," in *Workshop International sur Les Reseaux de Capteurs sans Fil en conjonction avec la 7eme Conference Internationale sur les NOuvelles TEchnologies de la REpartition (NOTERE), Marrakech, MAR, 4 juin 2007*, 2007, p. 5.

[29] "Net-SNMP," http://www.net-snmp.org/, 2007. [Online]. Available: http://www.net-snmp.org/

[30] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of ipv6 packets over ieee 802.15.4 networks," vol. RFC4944, 2007, pp. 1–30.

[31] K. M. Hou, J.-J. Li, and H.-Y. Zhou, "Cooperative inter-vehicle communication protocol dedicated to intelligent transport systems (CIVIC)," in *Journal of Harbin Institute of Technology*, vol. 39, 2007, pp. 145–152.