Simulation and visualization method of Wireless Sensor Network performances

T. Antoine-Santoni[1], J.F. Santucci[1], E. De Gentili[1], B. Costa[1]
[1]University of Corsica - UMR CNRS 6134 Quartier Grossetti, BP 52 - 20250 Corte - FRANCE
{antoine-santoni, santucci, gentili, bcosta}@univ-corse.fr

***Abstract*— The wireless distributed microsensor networks profit of recent technological advances and it seems essential to understand precisely these systems. Modeling and simulation appear like an essential aspect to predict the Wireless Sensor Network specific behavior under different conditions. We want to provide a new approach of modeling, simulation and visualization of Wireless Sensor Network using a discrete event approach. Described by Zeigler in the 70 's, the Discrete Event system Specification is ideal to describe the asynchronous nature of the events occuring in WSN. We try to provide a basis model to analyze WSN performance, as routing management, energy cunsumption or relative CPU activity. Our approach use a detailled definition of node oriented components and it wants to bring some ways to visualize the network at different level of abstraction.**

***Index Terms*— Modeling, Simulation,Visualization,DEVS, WSN, performance.**

I. Introduction

Advances in hardware technology and engineering design have led to reductions in size, power consumption, and cost. This has enabled compact, autonomous nodes, each containing one or more sensors, computation and communication capabilities, and a power supply. Networks of wireless sensors are the result of rapid convergence of three key technologies[1], [2]:

- Computing/Internet : computing power is becoming small and inexpensive enough to add to almost any object. networks of computers facilate collaboration trough information and resoure sharing
- Sensor : miniaturization, micromachining and low cost leads to smaller sizes, low power, lower costs. Alows to monitor with higher granularity. many types of sensors and more on the way
- Wireless/ Antennas : Spans a host ot technologies including Bluetooth and WiFi networks, cellular and satellite communications.

The sensor nodes are usually scattered in a sensor field [3] as shown in Figure 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink. Data are routed back to the sink by a multihop infrastructureless architecture through the sink. The sink may communicate with the task manager node via Internet or satellite. The design of the sensor network as described by Figure 1 is influenced by many factors, including fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media, and power consumption.

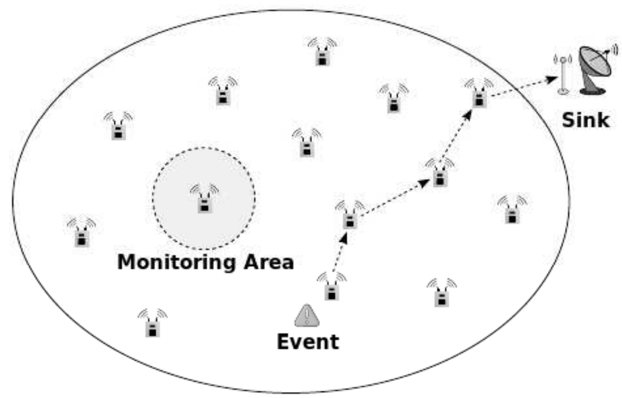

Fig. 1. Sensor nodes in a sensor field

A sensor node combines the abilities to compute, to communicate and to sense[4]. In a sensor network, different functionalities can be associated with the sensor nodes [5]. In earlier works, all sensor nodes are assumed to be homogenous, having equal capacity in terms of computation, communication and power. However, depending on the application a node can be dedicated to a particular special function such as relaying, aggregation. This work wants to provide some results of WSN capacity to detect Wildfire birth conditions. The rate of failure, the time of transmission data, the evolutionary architecture are blur concepts, which are not verifiable with such experiments. It allows a component-based approach for the design of complex systems. A discrete event approach like DEVS allows the modeling of the dynamics of the system based on the state, space, measured in a qualitative and quantitative manner, and a continuous time scale.

The goal of sensor is to send such collected data, usually via radio transmitter, to a command center (sink or Base Station) either directly or through a data concentration center (a gateway)

Based of node description in [6][3], the main components of sensor consist of a sensing unit, a processing unit, a transceiver, and a power unit as shown in Figure 2.

1) The communication tool is represented by the antenna and its role is to send some information on the channel;
2) Memory is the unit for storage action of information

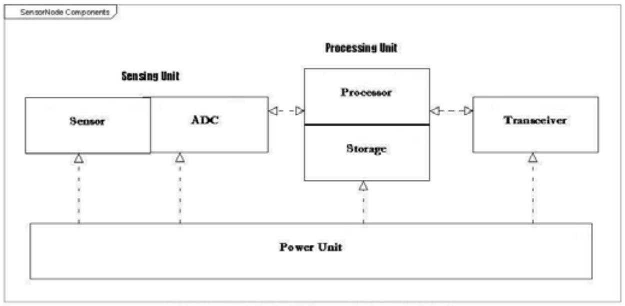

Fig. 2. Components of a sensor node

evolving in the node. Information have two sources : information coming from an other node and information coming from sensoboard in environmental monitoring case. In these both conditions, information are treated by processor ;
3) Processor treats all information in the node. CPU manages activity in the mote and reacts according to instruction type.
4) Battery defines lifetime of a node. Each component according to realized action consumes some energy. Energy consumption exists also in sleep state.
5) Sensor board regroups monitoring activities. It can transmit some information collected by sensor but also transmits message alert if some critics thresholds are reached.

Modeling and simulation appear like an essential aspect to understand the specific behavior of Wireless Sensor Network under specific conditions. The network simulation for sensors is a challenging problem as it has faithfully to model the constraints hardware and energy, which is typical with sensor nodes and also have to model various aspects exclusive to sensor networks. The hierarchical nature of DEVS makes it perfect for describing a system like sensor mote. The discrete-event nature improves the execution performance of a model like this due to the asynchronous nature of the events occurring in WSN. Some works exist for the modelling of Wireless ad-hoc networks using DEVS. In [7] , we can see the using of Cell-DEVS to model routing protocol AODV. In this paper, DEVS is used to formally specify discrete events systems using modular description. This strategy allows the reuse of tested models, improving the safety of the simulations and allowing reducing of development time. As it is discrete event formalism, it uses a continuous time base, which allows accurate timing representation, and reduces CPU time requirements. However this work focus on wireless network and it's strongly different like Wireless Sensor Network. In [8], a coupling between the NS-2 simulator (Ns [9],also popularly called ns-2, in reference to its current generation, is a discrete event network simulator) and DEVS definition is showed. We observe the behavior of a sensor node's application and its environmental behaviors such as battle fields which are defined in DEVS modeling and the roles of networking protocol behaviors which are assigned to NS-2 since NS-2 has well-designed network protocol libraries. However modular aspect of all components doesn't exist and seems no easy to implement environmental scenario. On the basis of this report, we make the choice to define all components of Wireless Sensor Network using DEVS formalism.

The paper is organized as follows : Section 2 introduces brievely the Wireless Sensor Network area. In Section 3 we present DEVS formalism with an relative overwiew of routing protocol. Section 4 shows our appraoch based on DEVS descripion of Wireless Sensor node. The results of simulation are presented in Section 4. Finally, in Section 5 we give some conclusions and directions of our future research works.

II. Overview of WSN Simulator

Nowadays, Wireless Sensor Network research has different focus and several fields of application: channel access control, routing protocol definition, network management, QoS, energy consumption or CPU activity. We can find different simulators to represent activity and performance of WSN.

SensorSim [10]extends the ns-2 network simulator with models of sensor channels, accurate battery and power consumption. Each node has a sensor stacks that acts as a sink to the signals in the sensor channels, accurate battery and power consumption.

Atemu [11] is a software emulator for AVR processor based systems. Along with support for the AVR processor, it also includes support for other peripheral devices on the MICA2 sensor node platform such as the radio. Atemu can be used to perform high fidelity large scale sensor network emulation studies in a controlled environment. Though the current release only includes support for MICA2 hardware, it can be easily extended to include other sensor node platforms. It allows for the use of heterogeneous sensor nodes in the same sensor network. Atemu can't represent different activities of hardware components because it use an high abstraction level.

TOSSIM [12]and PowerTOSSIM [13] are two important simulators which can describe correctly routing protocol, node applications or energy consumption but they are strongly dependents of TinyOS and can't represent generic framework for heterogeneous platforms.

In [14], Glonemo can be considered like a close approach of our work. Indeed, Glonemo bring some solutions as the MAC layer for description of Wireless sensor node however certain parameters appear uncertain as CPU activity, general energy consumption, sensing activity.

NAB [15]is a network simulator targeted at wireless ad hoc and sensor networks, written in ML which can describe some characteristics of WSN but appears as not suitable for an accurate modeling of energy.

SENS [16] is an application-oriented wireless sensor network which models ad-hoc static nodes. It provides models for a limited set of sensors, actuators, a model for the environment and a framework for testing applications. SENS appears like a suitable WSN simulator however some characteristics like addition of new models of sensor, modeling arbitrary ubiquitous computing environments.

It is particularly difficult to find simulator or model able to represent behavior of a node and able to generate particular environmental scenario. The need is to have a simulator able to represent sensor node at different abstraction level, which be able to describe components behavior in particular conditions. Modular aspects of components in sensor model don't exist clearly. Our work wants to focus on representation capacity of DEVS formalism. This possibility to distinguish different abstraction level is clearly essential to have definition of components activity of node on the one hand and general behavior in the network on the other hand.

III. DEVS FORMALISM

Based on systems theory, DEVS formalism was introduced by Professor B.P. Zeigler in the late 70s [17], [**?**]. It allows a hierarchical and modular way to model the discrete event systems. A system (or model) is called modular if it possesses the input and output ports permitting interaction with its outside environment. In DEVS, a model is seen as a "black box"S which receives and broadcasts messages on its input and output ports. DEVS defines two kinds of models: atomic models and coupled models, representing respectively the behavior and the internal structure of a part of a model.

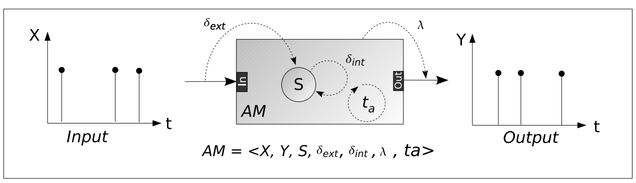

Fig. 3. DEVS atomic model

Figure 3 represents an *AM* atomic model with its output data Y calculated according to input data X. The *AM* atomic model has a state variable S that can be reached during the simulation. The functions δ_{ext}, λ, δ_{int} and t_a respectively allow the model's change of state when an external event occurs on one of those outputs (external transition function), the disposal of the output Y (output function), the model's change of state after having given an output (internal transition function) and finally the determination of the duration of the model's state (time advance function).

The coupled models are defined by a set of sub-models (atomic and/or coupled) and express the internal structure of the system's sub-parts thanks to the coupling definition between the sub-models.

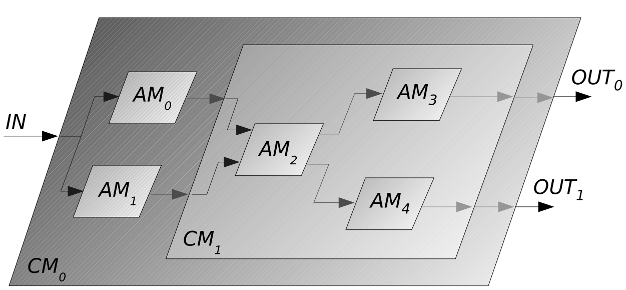

Fig. 4. DEVS coupled model

Figure 4 shows an example of the hierarchical structure of coupled model CM_0 which has an input port *IN* and two output ports OUT_0 and OUT_1. It contains the atomic sub-models AM_0, AM_1 and also the coupled model CM_1. The latter can encapsulate other models such as atomic models AM_2, AM_3 and AM_4. A coupled model is specified through the list of its components (AM_0, AM_1,AM_2, AM_3, AM_4 and CM_1), the list of its internal couplings ($AM_0 \rightarrow CM_1$ and $AM_1 \rightarrow CM_1$), the list of the external input couplings ($IN \rightarrow AM_0$ and $IN \rightarrow AM_1$), the list of the external output couplings ($CM_1 \rightarrow OUT_0$ and $CM_1 \rightarrow OUT_1$) and the list of the sub-model's influence ($CM_1 = \{AM_0, AM_1\}$ or CM_1 and influenced by AM_0 and AM_1).

DEVS formalism is mainly used for the description of discrete event systems. It constitutes a powerful modeling and simulation tool permitting a system modeling on several levels of description as well as the definition of the models' behaviors. One of DEVS formalism's important properties is that it automatically provides a simulator for each model. DEVS establishes a distinction between a system modeling and a system simulation so as any model can be simulated without the need for a specific simulator to be implemented. Each atomic model is associated with a simulator in charge of managing the component's behavior and each coupled model is associated with a coordinator in charge of the time synchronization of underlying components.

IV. MODELING METHOD

It seems important to us to represent the different basics hardware components of the node. Our generic approach leads us to define behaviour of different components. We try to delimit the different reaction of the node units to move towards the description of a general behaviour of a sensor using a discrete event formalism DEVS. A discrete event approach like DEVS formalism allows modelling of the dynamics of the system based on the state, space, measured in a qualitative and quantitative manner, and a continuous time scale. The advantages of this formalism for description of complex system in discrete-event scale appear clearly in number field of research however definition of sensor network and in particular sensor node don't exist. As sensor networks gaim more importance in the research communities, it's very crucial to show the advantages of DEVS formalism and to have a simulator with a modular structure. The use of this formalism in accordance with its definition implies for this research area two essential points: a modelling specification step and consequently a clear interpretation of simulations results in the real world and a non ambiguous operational semantic step allowing the introduction of a formal specification of mechanics of simulation using an abstract simulator.

In this part, we will present initially the structure of message exchanged between the nodes and also between components of one node. After we will introduce the different atomic and coupled models. These models are based of approach of node components shown on Figure 2.

A. Message description

1) Structure of message: We presents structure of our message:

- **Origin** defines the node which is the source of this message. Parent, this field is one of characteristic of reliable route protocol. It determines the node nearest to the basic station, the highest in routing table.
- **Sender** defines the node which sent this message.
- **Destination** defines the destination of message which has been treated by a node. The destination can be the sink or an other node.
- **Ndid**, node can be identified by a nodeID which correspond at an identifier of a node group.
- **Type**, this field is very essential because it will define the action of the different components of the system.
- **Hop,** this field appears in our message because it is a parameter of reliable route protocol. However it can be used for other routing protocol.
- **Link** appears also like a attribute of reliableRoute protocol. It indicates the quality of connectivity between two nodes and is very important for the definition of routing table.
- **Data :** *Temp* indicates temperature parameter coming from Sensor board, *Humidity, Pressure, GPS...*

Conso is a special internal message for energy cunsumption. For each action realized in the node, a message is sanded towards Battery with information about energy characteristics.

2) Message Type: Atomic Model can manage different types of message. According type of message, different models are able to do an action.

We can find different type of message and short description. This list is not complete but we can find the most important message.

- ***Router*** Message for AM Net : routing information
- ***BSCollect*** Message for AM Sensorboard : collect environmental data
- ***MemCollect*** Message for Memory : collect stocked Data
- ***ACK*** Message for Net : acknowledgement by a node
- ***Update*** Message for Net : routing table Update
- ***WhiteFlag*** Message for Net : network architecture signal
- ***DEAD*** Message for All Models : no energy in the node
- ***Alert*** Message from sensorboard : threshold reached

3) Atomic Model COM: AM COM is an atomic model for representation of communication in a node. The goal is to address message towards good nodes according to routing table in CM Process. On Figure 5, we represent only a link on input port or on output port with a different sensor node ; however it's possible to have more links depending on number of nodes connected. We define different states:

- Receipt for message coming on Inport1 from a node or Base Station (BS),
- Transmit for message going out sensor to an other node or BS, Busy is state of transition when a message is treated by MC processor,
- Free when there is no activity in node (node listen the channel),
- Dead when there is no battery in sensor.

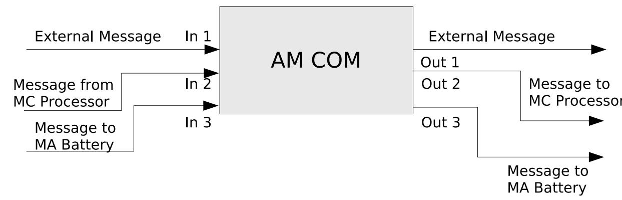

Fig. 5. Atomic Model COM

4) Coupled Model Process: CM Process is shown on Figure 6. It is the most important model of our approach. All messages coming from AM COM or AM Sensorboard are treated obligatorily by this model to have routing information. AM Processor manages all messages and all components. It is difficult to represent all action of processor but we try to bring some solution by a generic approach. Our CM Process can represent a simplest representation of a generic Operating System. Indeed we make the choice to decompose Processor in three Atomic model: AM Processor which can represent action management of OS and Processor, AM Net which manage the Network aspect and AM Flash which is a space to stock information but also the flash memory of node. These three models have only three states: Busy when model is in action, Free in sleep mode and Dead when there is not enough energy in the node. When a message comes, AM Processor sends it to AM Net that is a model of routing management. AM Net changes gives routing informations at the message. AM Flash is a simple atomic model which can stock some information, not environmental but coming from BS, by example new node ID.

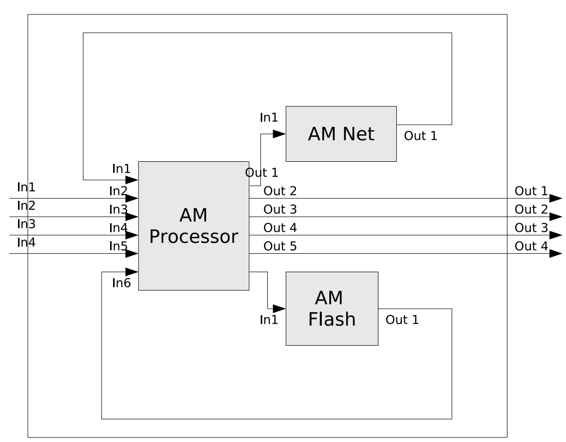

Fig. 6. Coupled Model Process

AM Processor treats all messages in the model and we can exprim relative activity of node by count of each action done by AM Processor.

For this model we use the characteristics of reliable route protocol. Indeed, estimation of link quality used to define neighbors table [18]. Link estimation is an internal value which changes according to a random definition of time. According to reliable route protocol, called Xmesh Protocol, one node sends its information to node with the most reliability of link. The mote in receipt phase is able to insert, evict or reinforce of neighbors table. It keeps a frequency count for each entry in the table. With this algorithm, a new neighbor is inserted in the table if there is an entry with a count of zero ; otherwise the count of all entries is decremented by one and the new candidate is dropped.

5) Atomic Model Battery : AM Battery as illustrated on Figure 8 is an atomic model connected to all models representing the components of the sensor. Each time there is an action using some energy, AM COM, CM Process and AM Sensorboard send a message to AM Battery.

For representation of enrgy cunsumption, we use for these first experiments a linear mode based on [19]. In linear model, the battery is treated as linear storage of current. The maximum capacity of the battery is achieved regardless of what the discharge rate is. The simple battery models allow user to see the efficiency of the user's application by providing how much capcity is consumed by the user. The remaining capacity C after operation duration of time t_d can be expressed by the following equation :

$$C = C' - \int_{t=t_0}^{t_0+t_d} I(t)dt \, , Eq.(1)$$

where *C'* is the previous capacity and I(t) is the instantaneous current consumed by the circuit at time t. The linear model assumes that *I(t)* will stay the same for the duration t_d, if the operation node of the circuit does not change for the duration t_d.

When a special value called size reached 0, dead message is sent for all components and all models adopt DEAD phase. All input ports are blocked and it is impossible for all models to change their state. Let us precise that all models have common important state called DEAD phase. When a sensor model enters in this special phase, it cannot act any more in the network. This particularity is essential for networking management.

6) Atomic Model Memory: AM Memory is a simple atomic model as shown on Figure 8, that allows storage of environmental data by CM Process. However, CM Process can have different actions on AM Memory according type of message. MemCollect and StockData are two types of message enough explicit to represent action of Processor.

7) Atomic Model SensorBoard: The goal AM Sensorboard is representing interactions between environment and sensors. It is an imporant point of our approach. AM Sensorboard is connected with a special atomic model that send data of environment AM Env. AM Env is an external model as shown on Figure 7 using environmental message to communicate with sensorboard. This interconnection between these two models represents sensing action of nodes in an environment or a specific phenomenon (wildfire). Am Env is able to send at periodical time different environmental values. To represent variation of environmental parameters, we have implemented a simple model where AM Env generate value at different step of simulation.

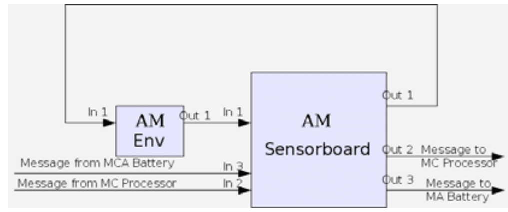

Fig. 7. Atomic Model SensorBoard

B. Coupled Model Sensor : definition of coupling

The model illustrated by Figure 8 represents fcoupling of our DEVS Coupled model of sensor. This definition is essential because it determines the connectivity between the model but also architecture characteristics of the future Wireless Sensor Network. Two input ports In1 and In2 and two output ports Out1 and Out2. In1 and Ou1 represent connectivity with a node and In2 and Out2 represent the connectivity with the environment. On Figure 8 appears the central role of our MC Process.

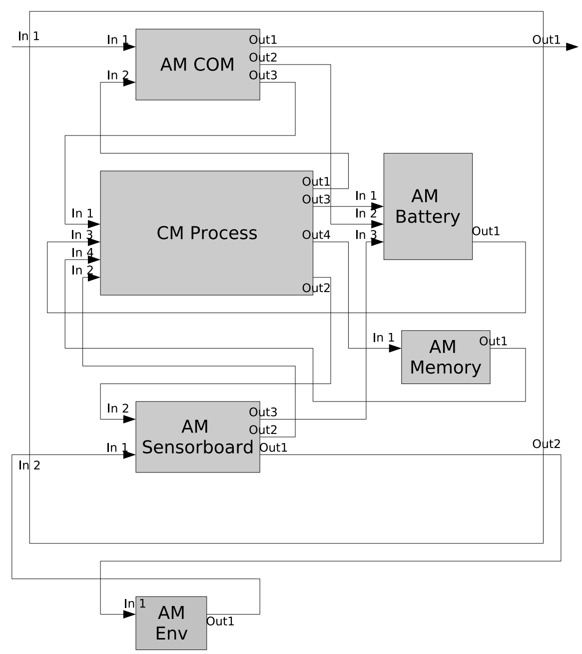

Fig. 8. Coupled Model Sensor

V. RESULTS

In Coupled Model Process, precisely Atomic Model Net, we work with a specific protocol, called Xmesh protocol or reliable route protocol [18]. This protocol allows the node to estimate the quality of the link from the other nodes passively by collecting statistics on packets it happens to hear, or by actively probing. Link quality is measured as the percent as of packets that arrived undamaged on a link. Link status and routing information are maintained in a neighborhood table. The goal is to have a neighborhood management algorithm that will keep a sufficient number of good neighbors in the table regardless of cell density. To maintain this routing table, this protocol use an algorithm based on a frequency count for each entry in the table. On insertion , a node is reinforced by incrementing its count. A new node will be inserted if there is an entry with a count of zero; otherwise the count of all entries is decremented by one and the new candidate is dropped. The neighbour table contains many fields : Group Ids, Parent node Ids, Chil Ids, reception link quality, link estimator data structures. To estimate link quality, Shortest Path protocol is used. For Shortest Path protocol, a node is a neighbor if its link quality exeeds threshold t. An another parameter is the selection of parent node. The cost metric is used to guide routing. The cost of a node is an abstract measure of distance ; it may be number of hops, expected number of transmissions, or estimate energy required to reach the sink. A neihgbor is selected as a potential parent only if its cost is less than the currrent cost of a node. This protocol is able to detect and avoid cycles, detect failures of transmision and eliminate node in the tree if link quality worsens.

VI. Visualization of Performance

On Figure 9, we represent the eight nodes and the BaseStation and we can see differents relation between the nodes. Theses relations represent connectivity and exprim capacity of communication between two nodes. If there is no connection between two nodes, it means that nodes are two much distant to exchange some information. During Simulation, all nodes send periodically messages towards BaseStation. Figure 9 shows only a predefinition of relations between nodes. We make the choice of this representation to work on routing protocol and representation capacity of our model. During simulation, we want that nodes make a choice according to routing protocol rules to reach the base station.

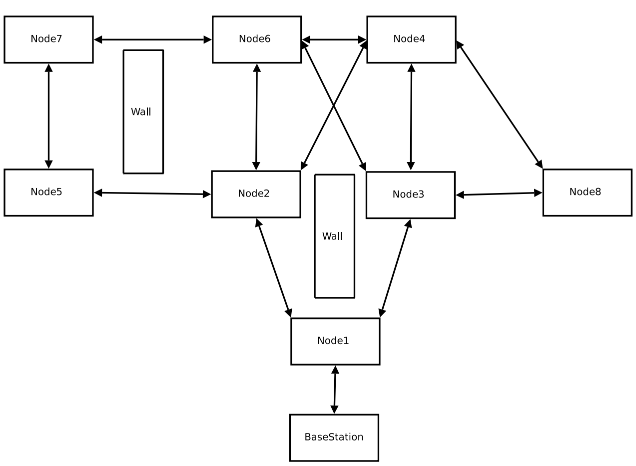

Fig. 9. Network architecture for simulation

A. Network Management

On Figure 10, we can see architecture of Wireless Sensor Network. This figure shows priviligied relations ,i.e. the first neighbor in routing table of each node according to the routing protocol.

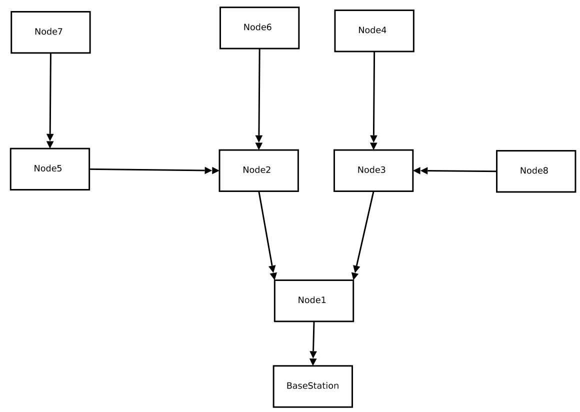

Fig. 10. WSN priviligied communication after 10 mn of simulation

On Figure 11, we can see the evolution of architecture of WSN. We can observe that the relations between sensors are differents. Indeed, this evolution means that routing table of Node6 has for first neighbor Node2 instead of Node3. This selection of the first neighbor is made by routing rules of Xmesh according to a goog link quality .

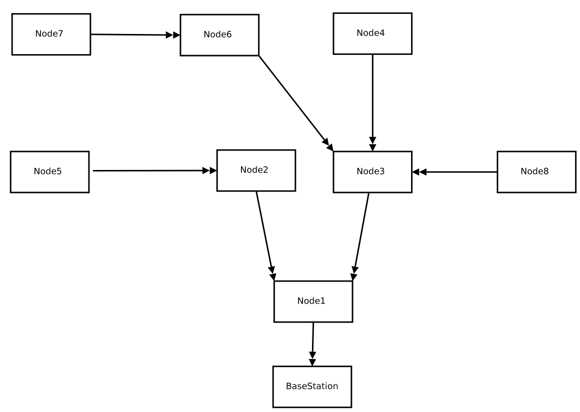

Fig. 11. WSN priviligied relation after 15 mn of simulation

On Figure 15, we have highlited time of apparition of each node on sink table that shows the difference between a node near the sink and node more distant. We can see on Figure 15 that Node7 appears after 80 seconds. This results confirm that distance between a node and Base Station has a great importance and alsoa real important shift due to no direct relation between nodes and BaseStation. The main explanation is the routing protocol. By example, message of Node7 needs

 ISBN # 1-56555-317-9

to pass through Node5, Node2, Node1 to reach Base Station, way defined by reliable route protocol.

On TABLE I, we can see the routing table of each node for 10 minutes of simulation. It appears for each node neighbors classification defined by rules of routing protocol. Indeed, the first neighbor for each is the node with minimum cost, i.e. node with the minimum distance in hop to reach BaseStation and with the maximum link quality. This table allows each node to transmit data information towards the sink with maximum security.

	Neighbor1	Neighbor2	Neighbor3	Neighbor4
Node1	BS	Node2	Node3	
Node2	Node1	Node4	Node5	Node6
Node3	Node1	Node4	Node6	Node8
Node4	Node3	Node2	Node6	Node8
Node5	Node2	Node7		
Node6	Node3	Node2	Node7	
Node7	Node6	Node5		
Node8	Node3	Node4		

TABLE I

ROUTING TABLE AFTER 10 MN OF SIMULATION

The nodes use for this classification the different parameters according to the rotuing protocols rules **: LinkQ** is link quality estimation according the variation that we can observe in WSN, **Frequency** represent the value based on Frequency algoritm, **Hop** determine number of hop to reach the sink for each node. For more precision, reliable route protocol define as parent of a node, the neighbor with the minimal cost in distance to reach the sink and with a good link quality. By example, we can observe differents neighbors of Node4 with classification according to reliable route protocol. In Node4 routing table, first neighbor is Node3. Indeed, this Node3 has a number of hop short to reach BaseStation but also a great quality of link. If Node4 wants to send some environmental informations towards the sink, it uses for first relay the Node3. This table evolve during simulation fucntion link quality variation.

B. Energy cusumption and CPU activity

Our approach allows us to distinguish relative energy and cunsumption and processor activity.

On Figure 12, we represent the events treated by each node processor model during 11 minutes. In our approach, AM Processor manages all the components and treated all action in the node. We can observe a important activity of Node 1 because it has a central role in the network and it is the bridge between the other nodes and the sink. This activity of Node 1 is the direct effect of the nodes deployment and network architecture chosen for the test represented on Figure 9.

The consequence of this network deployment is a different consumption for each node as shown on Figure 13. We can distinguish clearly the important energy cunsumption of Node 1. Let us precise that the results of energy cunsumption are based on linear model described in [19]. Our goal is just to represent the role of node deployement in power management.

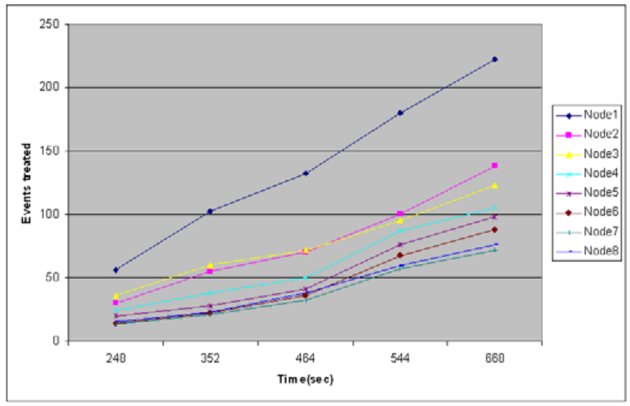

Fig. 12. Relative CPU activity in the WSN

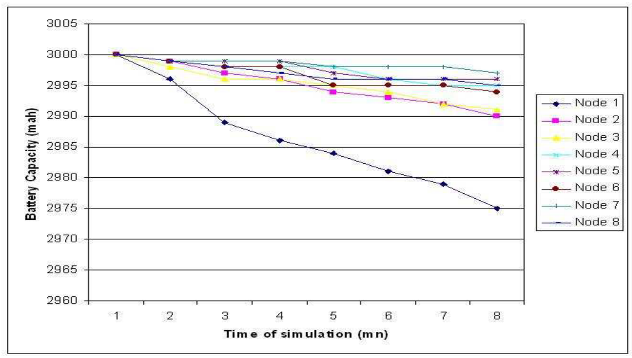

Fig. 13. Relative energy cunsumption in the WSN

C. Sensing activity

According to our approach, we implemented our AM Env with a simple temperature model with a rapid increase of temperature that it can be observed in wildfire case . On Figure 14, we analyse environmental data sended periodically by the nodes and we observe that the simple temperature model is clearly represented. We observe for each node a rapid increase of temperature .

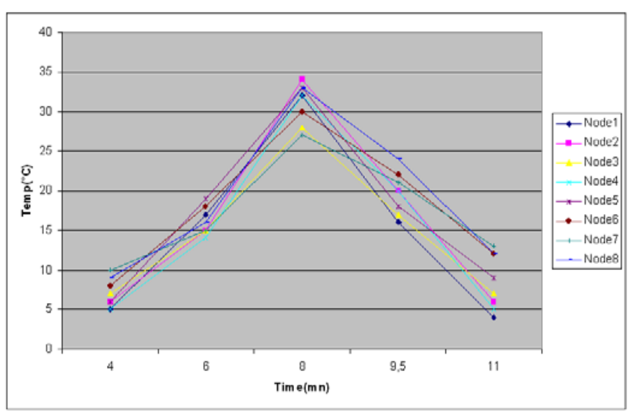

Fig. 14. Environmental Temperature parameter during simulation

D. Latency Time

On Figure 15, we have highlited time of apparition of each node on sink table that shows the difference between a node near the sink and node more distant. However, this time is not very important because we can see on Figure 15 that Node7 appears after 80 seconds. Figure 15 shows a real important shift due to no direct relation between nodes and BaseStation.

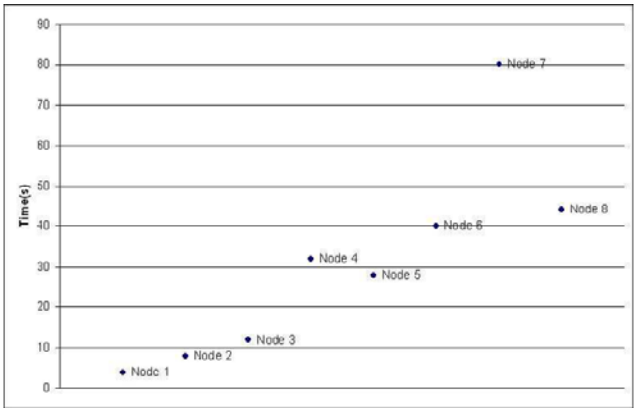

Fig. 15. First apparition of node message on the sink

The main explanation is the routing protocol. By example, message of Node7 needs to pass through Node5, Node2, Node1 to reach BaseStation, way defined by reliable route protocol.

VII. Conclusion and Future Work

This article provides a modeling and simulation of performance of a Wireless Sensor Network. This work is based on the DEVS formalism for the modelling and the simulation of complex discrete event system. We have demonstrated the capacity of our approach to analyze the evolution of Wireless Sensor Network architecture and analyse certains performance of WSN. According to our approach of modeling a sensor components, we provide results simulation of a Wireless Sensor Network with eight nodes. We have highlighted routing parameters, relative energy cunsumption and CPU activity and we showed the global characteristics of the system like the number of messages received by the sink. These results want to show that our first approach of WSN using DEVS formalism is good and provides a new level of visualization of node. Indeed, a DEVS description of components allows us to visualize the characteristics of each components. These results confirm a good approach of WSN characteristics but also highlight the advantages of DEVS formalism in this domain. Modular aspect of DEVS allows that each model of sensor can be changed by an other model. After the completion of the main components of the Sensor network an application to test the model can be created. This application is based on a DEVS simulator written in Pyhton developped by [20]. It is divided in four packages : package DEVS, ackage ComponentsNodes, package SimulationTools, and wireless sensor networ specification package. A simulation tool called DEVS-WSN is currently in developement phase.

References

[1] Crossbow-Technology, *Wireless Sensor Network Seminar*. Como, Italy.

[2] A. Hac, *Wireless Sensor Designs*. John Wiley and Sons Ltd, 2003.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag. 40*, vol. 8, pp. 102–114, 2002.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 263–270, ACM Press, 1999.

[5] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, 2002.

[6] I. Khemapech, I. Duncan, and A. Miller, "A survey of wireless sensor networks technology," in *In PGNET, Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2005.

[7] U. Farooq, B. Balya, and G. Wainer, "Modelling routing in wireless ad-hoc networks using cell-devs," in *In Proceedings of 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, (San Jose, Californie, USA), pp. 285–292, 2004.

[8] T. Kim, "Devs-ns2 environment : An integrated tool for efficent network modeling and simulation," Master's thesis, 2006.

[9] "The network simulator - ns-2.."

[10] S. Park, A. Savvides, and M. B. Srivastava, "Sensorsim: a simulation framework for sensor networks," in *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, (New York, NY, USA), pp. 104–111, ACM Press, 2000.

[11] J. Polley, D. Blazakis, J. Mcgee, D. Rusk, and J. S. Baras, "Atemu: a fine-grained sensor network simulator," in *Sensor and Ad Hoc Communications and Networks. IEEE SECON 2004*, pp. 145–152, 2004.

[12] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 126–137, ACM Press, 2003.

[13] V. Shnayder, M. Hempstead, B. rong Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 188–200, ACM Press, 2004.

[14] L. Samper, F. Maraninchi, L. Mounier, and L. Mandel, "Glonemo: global and accurate formal models for the analysis of ad-hoc sensor networks," in *InterSense '06: Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*, (New York, NY, USA), p. 3, ACM Press, 2006.

[15] EPFL, "Network in a box [online]."

[16] S. Sundresh, W. Kim, and G. Agha, "Sens: A sensor, environment and network simulator," in *ANSS '04: Proceedings of the 37th annual symposium on Simulation*, (Washington, DC, USA), p. 221, IEEE Computer Society, 2004.

[17] B. P. Zeigler, *Theory of Modeling and Simulation*. Academic Press, 1976.

[18] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys03: Proceedings of the 1st international conference on Embedded networked sensor systems*, (New York, USA), pp. 14–27, ACM Press, 2003.

[19] S. Park, A. Savvides, and M. Srivastava, "Battery capacity measurement and analysis using lithium coin cell battery," in *ISLPED '01: Proceedings of the 2001 international symposium on Low power electronics and design*, (New York, NY, USA), pp. 382–387, ACM Press, 2001.

[20] J.-S. Bolduc and H. Vangheluwe, "pythonDEVS : A modeling and simulation package for classical hierarchal DEVS," in *Technical Report, MSDL, University of McGill*, 2001.