ParrotNet: a Mechanism for Data Propagation in Highly-Dense Low-Mobility Wireless Sensor Networks

Carlos A. dos Reis Filho and Rogério Esteves Salustiano
School of Electrical and Computer Engineering (FEEC)
State University of Campinas (UNICAMP)
Campinas, Brazil
{carlos_reis, rsalusti}@lpm.fee.unicamp.br

Abstract – **This paper describes a novel wireless sensor network data communication strategy, which is appropriate for systems that feature high density of nodes, low mobility and pre-established lifetime. Another constraint featured by the targeted systems is the power of a node radio. The range that the transmitting radio can reach is below one meter radius and the messages it sends have a fixed length. In the targeted network structure to which the data communication strategy has been conceived, the nodes are limited to react against a stimulus performing one out of four possible tasks, which can be easily implemented by a state machine. The proposed data communication strategy has been exhaustibly tested with a network simulator, which adequately models all participants of the system and, in addition, it provides a real-time visual representation of the system status. Simulation results of a network comprising 30 nodes allocated within a bounded 2m radius circle changing their positions by random jumps of 5cm every 2s have accomplished 98.4% of success in data transmission.**

Keywords- wireless network; protocol; simulation

I. INTRODUCTION

The almost boundless scope of applications of Wireless Sensor Networks (WSN) has motivated significant research investments toward dominating the technology in its various aspects [1]. One of the biggest challenges, so far an unsolved problem, has been the lack of appropriate means of providing energy to sustain the network nodes active for long periods of time, without compromising volume, weight, appearance and environment damage as well. To date, the big majority of physically implemented WSN have been powered with batteries, mostly violating at least one of the rules that dictate what a WSN is expected to be [2]. Fortunately, techniques like energy scavenging and energy induction are rapidly evolving to commercially viable levels, reinforcing the actual practicability of WSN within the near future. It is important to notice, though, that while the scarceness of energy remains a fundamental limitation, the success of a WSN relies on the strategy that is devised to make it work. Hence, WSN are application specific systems [3].

This paper addresses a particular type of WSN that features a high density of nodes, which moves however within a short range, have a well-established lifetime and the range covered by its transmitting radio is very small. In operation, the data acquired by each sensor node must reach the base station, upon request of the data collecting base station. Contrary to what happens with the sensor nodes, the base station operates with unlimited energy, so that every message it sends reaches the entire network. The rate of data collection from each node by the base station is low, just a few times per day.

II. NETWORK FUNCTIONING

In the described scenario the base station is the only target address for the data either produced or acquired by each individual node. Consequently, a suitable communication network approach is a master-slave structure in which the base station plays the full-power master, while the nodes are energized by a battery that must last not less than the time predicted for the existence of the network.

In the conceived data communication strategy, a message sent by the master can be listened by all nodes in the network, although it is always addressed to individual nodes. The master communicates with the slaves by sending two different messages: One message commands the addressed node to transmit the data that it can either produce or acquire at that moment. The other message, which is also addressed to a specific node, commands that node to transmit the data that it is holding in its memory buffer at that moment.

The sensor nodes have limited processing power, the only embedded memory is a single buffer restricted to a few-bytes register and the range of the transmitting radio is limited to tens of centimeter to a meter at most.

A sensor node comes into action for a very short time. Most of the time, it is at rest consuming only the energy required to keep the receiving radio in alert mode (SLEEP). It wakes up whenever a radio broadcast occurs. Upon waking up, a sensor node decodes the receiving message in order to recognize both the addressee and the commanded task. As a result of the message decoding process, the node executes one of the programmed tasks, which can be explained with the help of the flow diagram in Fig. 1.

One type of commanded task ordered by the master is "REPORT DATA". This message starts a new scanning node round. Whenever this command is received by the addressee of the message, the sensor node codes and packs the data it has produced or acquired at that moment, stores it in its memory buffer, sets a buffer flag to indicate that the buffer contains a valid data, broadcasts a message with this information and eventually returns to sleep. When the node that receives this message is not the addressee, it simply clears the buffer flag and enters in SLEEP mode.

This work has been supported by National Council for Scientific and Technological Development (CNPq - Brazil) and Brazilian Agency for Personnel Development (CAPES - Brazil).

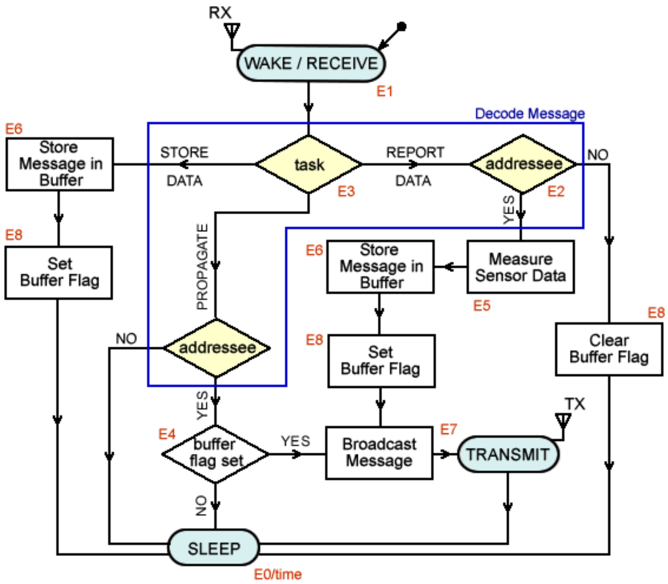

Figure 1. Sensor node functioning flow diagram.

Another type of commanded task is "PROPAGATE". Whenever it is received by the addressee node, which has a valid data in the buffer (buffer flag is set) it acts like a "parrot", repeating the last heard message. Actually, it simply broadcasts the contents of its memory buffer.

The third type of task that a node can execute upon receiving a message, "STORE DATA", occurs when the message has been sent by another node. In this case, the listening node stores the receiving message in its memory buffer and returns to sleep.

Due to being highly dense and because the sensor nodes move within a relatively small area, whenever a node broadcasts a message it is received by a number of close neighbors, which will be individually ordered to propagate the received information. As a result, after successive commands from the master, the message eventually reaches the collecting base. This is the mechanism by means of which the base station collects data from the nodes in the network.

Fig. 2 shows an example of a virtual path created in the delivery of data from sensor node SN1. The base station BS ordered SN1 to "REPORT DATA" and successively ordered SN7, SN5 and SN2 to "PROPAGATE. When sensor node SN2 sent "STORE DATA", after receiving "PROPAGATE", BS finally received the data from SN1. Note that SN9, SN17, SN18 and SN13 also have the data from SN1 in their buffers.

III. NODE-ROUND AND GLOBAL-ROUND

The concepts of node-round and global-round have been used to facilitate the process by means of which the base station retrieves data from the sensor nodes. A node-round starts with a command "REPORT DATA" addressed to an specific node. Following this command, the base station selects randomly one of the other nodes and commands it to propagate the contents of its memory buffer. Notice that this task will be effectively executed only if the buffer of the commanded node contains a valid data.

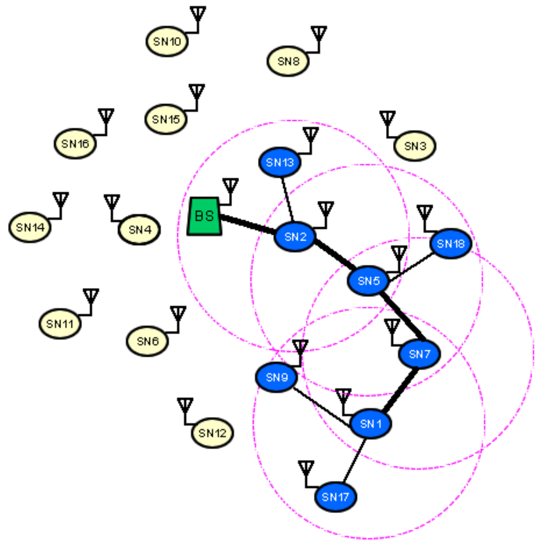

Figure 2. ParrotNet virtual path example: the delivery of data from sensor node SN1 to base station (BS).

Every time the base station sends a "PROPAGATE" it expects to listen a coming message during a finite time. If no message is received during this time, a timeout occurs and a new "PROPAGATE" is sent to another node, also randomly selected. When all nodes in the network, excluding the node commanded to report its data, have been addressed to "PROPAGATE" and no message arrived at the base station, all nodes in the network will be commanded a "PROPAGATE" one more time, following a criteria of random selection. This process can be repeated several times.

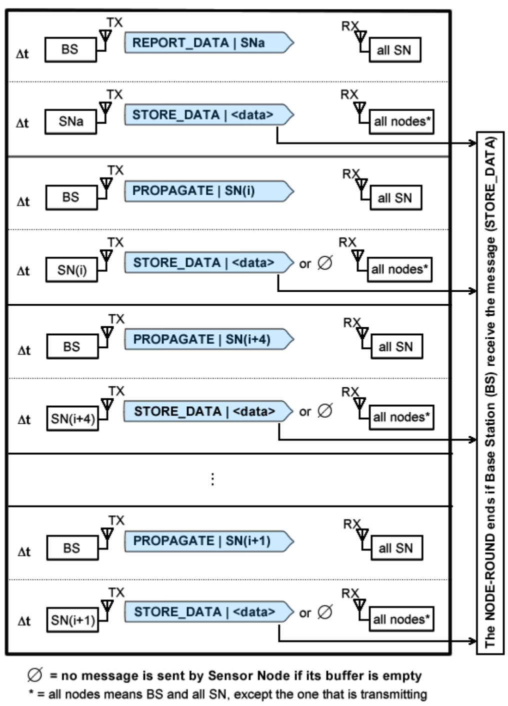

Figure 3. Node-Round flow for sensor node SNa.

A node round is completed either when the message eventually arrives or when the number of trials has been achieved. In this last case, the data could not be retrieved. An illustration of this process is shown in Fig. 3. A global-round, in turn, is completed when all nodes in the network have been commanded a "REPORT DATA".

IV. NETWORK ENERGY MONITORING

The worst-case battery capacity to sustain the slave node active during the network lifetime is estimated on basis of the knowledge of the required amount of energy for executing each one of the types of task that the node executes, as well as the number of times that each one of the tasks will be executed.

Since the node implements a state machine with a finite and well-established number of possible flow paths, the energy cost of each type of task can be easily calculated. Table I relates every individual action that the node executes with the corresponding energy cost.

TABLE I. NODE ACTION VERSUS ENERGY COST

ACTION	ENERGY COST
SLEEP	E_0/unit time
WAKE/RECEIVE	E1
Recognize address	E2
Recognize task	E3
Recognize flag	E4
Measure Sensor Data	E5
Store Message in Buffer	E6
Broadcast Message/TRANSMIT	E7
Set/Clear Buffer Flag	E8

Once the energy cost per action is known, an estimation of the energy cost to execute a task can be straightforwardly accomplished. For instance, referring to both Fig. 1 and Table I, when a node responds to a "REPORT DATA" command, an amount of (E1+E2+E3+E5+E6+E7+E8) will be decremented from its available energy. When the node responds to a "PROPAGATE" command from the master, and it is not the addressee, the spent energy equals (E1+E2+E3+E4). Otherwise, in case it is the addressed node, that amount of spent energy will be increased by E7. The energy cost that corresponds to the task of storing in the buffer the message that the node listen from a neighbor transmitting node is (E1+E3+E6+E8).

In addition to the momentary expenditure of energy, every node features a continuo energy loss due to the permanent state of alert of the receiving radio. The energy cost of the state of alert (SLEEP) is a function of time

Instead of selecting the nodes to propagate data following a numerical order, it was preferred to select them randomly as a means of energy saving. Notice that if a numerical order is adopted, the first nodes in the line will always be commanded to participate, thus being more prone to a shorter lifetime.

Another reason for the random selection criterion is the greater probability of success in data retrieval, since a change in the order of propagation means to enable new paths. Thus, both node-round and global-round use random sequences of sensor nodes to demand tasks.

V. PARROTNET SIMULATION

The proposed ParrotNet was simulated using the network simulator described in [4]. Developed in JAVA language, the simulator allows a detailed description of the network components, including hardware and communication protocol. In addition, it implements an energy monitoring mechanism which results in a log file containing the history of power consumption of every sensing node.

Fig. 4 illustrates a simplified class diagram of the used simulator. Class *ParrotNet* generates the network nodes and starts simulation. Classes *SynchGlobal*, *SynchThread*, *SynchRandom* e *SynchLogger,* are parts of the core of the simulator an allow the time coordination of threads. A detailed description of these classes is given in [4].

A network node can be either a *SensorNode* or a *BaseStation*. A *SensorNode* comprises a *Battery*, whose charge is gradually reduced, and a sensor, which provides the data that the sensor node transmits when commanded to. Both node types *BaseStation* and *SensorNode* have a *Radio*, which allows the nodes to communicate over the *Medium*. The nodes also have a physical position determined by the *MediumPosition*. Sensor Nodes changes, at an established rate, their positions in the *MediumPosition* class to perform movements in the *Medium*.

Classes *NetLogger, GlobalRoundLog* and *NodeRoundLog* are used to log the traffic of messages in the network as well as the status of the nodes. Classes *NetLogger, GlobalRoundLog* and *NodeRoundLog* are used to log the traffic of messages in the network as well as the status of the nodes. The class diagram of Fig. 4 does not include the interface classes.

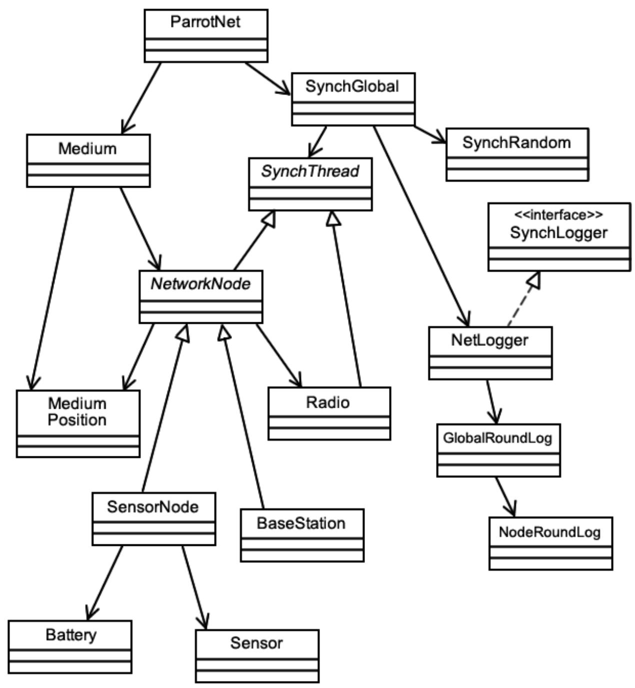

Figure 4. ParrotNet Class Diagram for Simulation

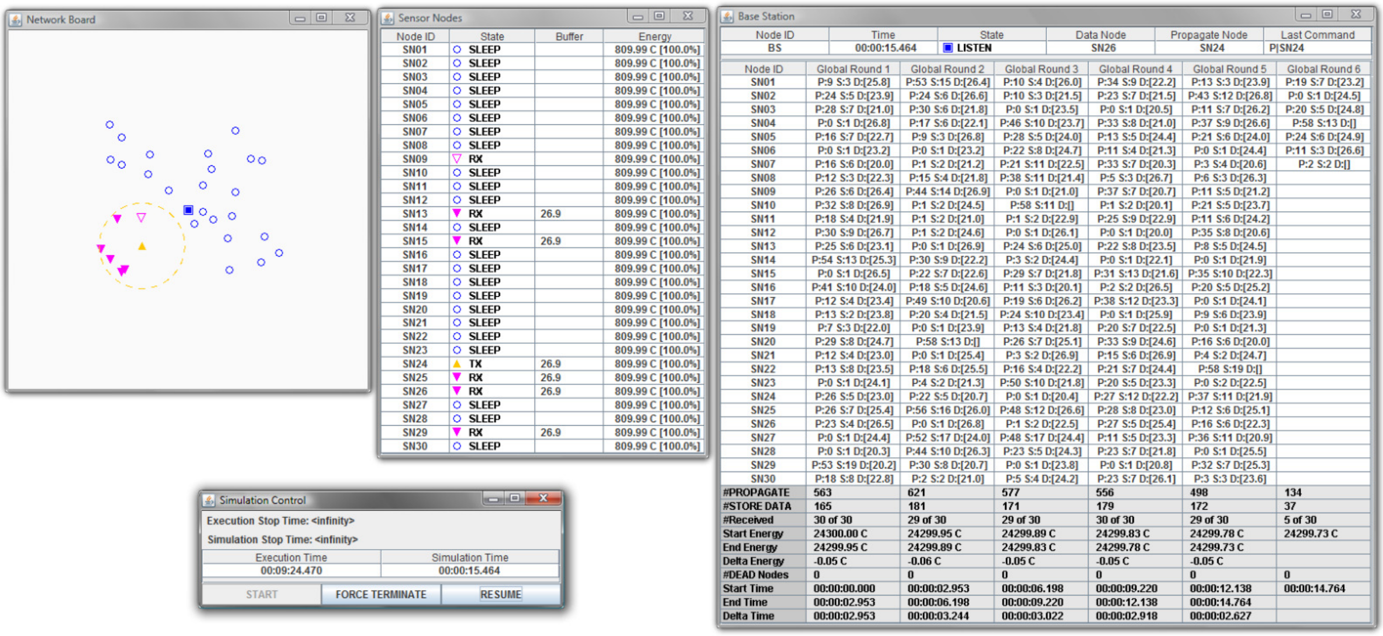

Figure 5. ParrotNet simulator graphical interface. Screen snapshot showing results of five global-rounds and parte of the sixth global round.

The simulated network comprises 30 sensor nodes, randomly located around the base station, occupying a region that is bounded by a 2.0m radius circle centered at the base station. The TX/RX radio associated with each sensor node operates with a transmit/receive rate of 500kbps and features the specifications of a commercially available radio [5]. The amount of energy that the radio consumes depends on the function it performs, as summarized in Table II. All digital signal processing is performed by a MCU [6], whose clock frequency is 1MHz. The relevant specifications regarding the power consumption of the used MCU are also shown in Table II.

TABLE II. ENERGY CONSUPTIOM OF SENSOR NODES OPERATING IN DIFFERENT STATES

DEVICE	STATE	CURRENT CONSUPTIOM
Radio	SLEEP (with WOR)	900 nA
	TX	21.2 mA
	RX	19.6 mA
MCU	IDLE	0.35 mA
	POWER DOWN	1 µA
	ACTIVE	1.1 mA

Other parameters used in simulation are listed below.

- Base station TX range: Infinite
- Sensor node TX range: 0.5m
- Sensor node initial energy: 225mAh
- Maximum number of node-rounds per node in a global-round: 2

Several simulations were carried out changing the speed with which the nodes move within the network area. A random number generator is used to find the new position of each node (coordinates within a bounding 2.0m circle), whenever a new location map is generated. The mechanism adopted to simulate the dislocation of the nodes uses a fixed dislocation length of 5cm while the map refreshing time is the changing variable. Hence, a shorter refreshing time means a faster move of the nodes.

In the simulation approach used to evaluate the functioning of ParrotNet, three graphical interfaces were developed. One of them, called Network Board, shows the momentary physical allocation of every sensing node. The other, called Sensor Nodes, shows the current status of every node, along with the status of its respective buffer (if empty or with valid data) and the remaining amount of energy. The third developed interface, called Base Station, shows the most recent action of the base station interacting with the sensor nodes, providing details like what order has been given and to which node it is addressed.

The Base Station graphical interface also informs how many nodes have been addressed a “PROPAGATE” command and how many, among them, did have data in the buffer to execute the received task. In addition, it shows the counting result of the number of data, which has arrived at the base station (indicating success in the establishment of virtual paths). The start time and end time of each global-round is also shown at the Base Station interface.

A further set of information presented at the Base Station graphical interface includes the remaining energy per node at the end of every node-round, the total remaining energy (sum of all nodes) and the number of dead nodes. It is important to mention that these data are only available at the simulation level. The network protocol does not provide means to allow these data to reach the base station.

An illustration of the developed graphical interfaces is shown in Fig. 5. It also shows the simulation control, which indicates the simulation execution time and clocked simulation time.

VI. CONCLUSIONS

A novel data communication mechanism for networks that feature high density of nodes, low mobility and pre-established lifetime has been devised and evaluated at the simulation level. Results from extensive simulations, changing several properties of the comprising components have shown that the proposed concept is functional and satisfactory in a myriad of applications. The developed approach is based on a master-slave network structure with slave nodes featuring limited energy resources. The successful delivery of a data requested by the full-power master to a slave node up to its final destination is dependent upon the status of several variables in the network, whose momentary physical arrangement is practically unpredictable. Some arrangements are favorable to the establishment of paths for the propagation of data until the base station, others not. For this reason, successive trials have to be performed. Simulations of a large number of different network configurations have shown that the index of success is above 95%.

REFERENCES

[1] Ammar, M. et al, "Report of the National Science Foundation Workshop on Fundamental Research in Networking", in Report of the National Science Foundation Directorate for Computer and Information Science and Engeneering (CISE), Virginia, April 24-25, 2003. <http://www.cs.virginia.edu/~jorg/workwhop/NSF-NetWorkshop-2003.pdf>.

[2] M. Ilyas and I. Mahgoub, "Handbook of sensor network: compact wireless and wired sensing systems," Florida, CRC Press, 2004, 775p.

[3] RUIZ, Linnyer B. et al., "Arquiteturas para Redes de Sensores Sem Fio" Tutorial of the simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2004. <www.dcc.ufmg.br/~mmvieira/publications/04mc-sbrc.pdf>.

[4] Salustiano, R. E. and Reis Filho, C. A., "Barrier Synchronization Simulator for Wireless Sensor Networks," in Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, (ICCDCS 2008), 2008, Cancun. p. 1-5.

[5] CC2500 - 2.4GHz RF Transceiver for Low-Power Wireless Applications, Chipcon. <http://www.ti.com/lit/gpn/cc2500>.

[6] ATMega32L - 8-bit AVR Microcontroller with 32Kbytes In-System Programmable Flash <www.atmel.com/dyn/resources/prod_documents/doc2503.pdf>.