Self-Deploying Wireless Sensor Network

Predrag M. Ćirković, *School of Electrical Engineering, Belgrade, predrag.cirkovic@gmail.com*
Zoran Lukić, *DunavNET ltd., Novi Sad, zrn.lkc@gmail.com*

Abstract **— Self-Deploying Wireless Sensor Network (SD-WSN) is a mesh network, capable to deploy itself to a specified location by controlling its motors and using the location engine on its nodes. Its gateway node to WAN network is used to relay gathered sensor data to the remote server. Numerous types of sensors can be attached to the deploying sensor node e.g. Light, CO (Carbon monoxide), Propane and LNG (Liquefied Natural) gas, Metal detectors.**

Keywords **— 6LoWPAN, locating, self-deploying, wireless sensor network, WSN, sensors, location.**

I. Introduction

THIS paper presents an overview of the possible realization techniques, commonly employed algorithms and examples of applications for this type of network.

The first part of the work refers to the theoretical approach to forming and using a self-deployed network. The goal is to explain: how can we program a single node to get desired behavior of random number of similar nodes in random space distribution, which are connected to each other and create a mesh network. What kind of data can we get from such a network and how can we process and present them, considering that we have an access to some of the nodes and each node can access to its neighbors?

Many of these algorithms are derived from algorithms taken from DSP technique and it is planned to implement those algorithms using standard hardware platforms and protocols for wireless sensor networks.

The second part of the work refers to the implementation example of several nodes connected to a self-deploying network. Numerous types of sensors can be attached to the deploying sensor node e.g. Light, CO (Carbon monoxide), propane and LNG (Liquefied Natural) gas, Metal detectors.

Applications for this capability of WSN are great, for example when a large number of nodes are needed to be placed on predefined positions, when it is necessary for time of placement to be short, to take measurements that can be hazardous for humans, when it is needed for space distribution to be determined by certain measurements and its propagations (diffusion through the space) etc.

II. Self-Deploying Network Definitions

In the similar study [11], where the centralized and distributed localized algorithms have been considered, the authors have been focused on sensor deployment and coverage, routing and sensor fusion. In this case, the several definitions, used in the single node design, are established in order to produce the desired behavior of the whole mesh network composed of a large number of such nodes.

The Fig. 1 shows the distances' definitions and the three main scenarios and relationships between nodes in a mesh network:

1. *Visibility* is the distance beyond which the two nodes are not directly connected.
2. *Raster* is the regular distance at which the nodes should be moved.

The Fig. 1 shows the three main scenarios and relationships between the nodes (the central and the node in the focus) in the mesh network:

1. The nodes are at the regular distance and are able to communicate between each other,
2. The nodes are too close and should be moved at the regular distance between each other,
3. The nodes are not connected and there is no direct link between the nodes. They can communicate with each other only over the other nodes in their link-lists.

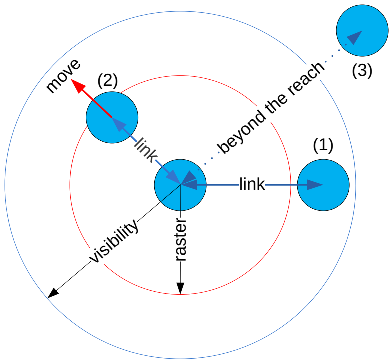

Fig. 1. Several scenarios for the nodes' positions.

III. Realization and Simulation

A. Node Realization

Each of the nodes has to have the several items in its attribute list:

1. *id* – unique identification number
2. *x, y, z* – node's coordinates in 3D space.
3. *temperature* – or some other measuring value.
4. *link_list*, – lists of nodes with established links (visible nodes).
5. *near_list*, *far_list* – collections of visible nodes inside the boundaries (*near* and *far* distances).
6. *direction_array* – coordinates of the vector of movement.

There are several worldwide project proposals in similar research fields. The authors would like to contribute to them with their ideas and experience. If someone in management is interested in cooperation or donation, shall feel free to contact the authors who will appreciate it.

7. *served_flag* – (used in simulation) in the real application the nodes will be timer-driven.
8. *path_list* – (used in simulation).

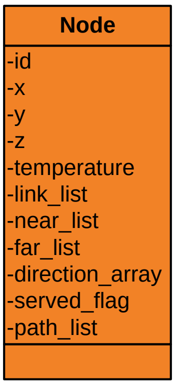

Fig. 2. Attributes for the class *Node* used in simulation.

A. *Calculating the Direction of Movement*

Each of the nodes in the network has to know its *current position* and the *positions of the other nodes* to calculate its *next position*.

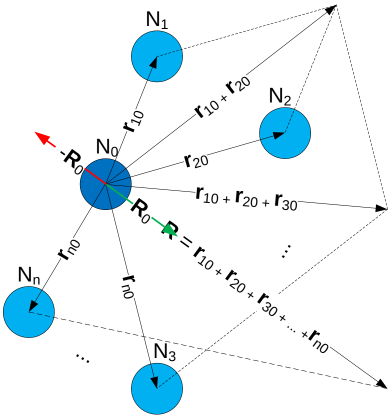

Fig. 3. Calculating the *fission* and *fusion* movement vectors.

Therefore, each node *pools* the other nodes in its link-list and requests coordinates form them. After the data reception, it (in the Fig. 3 node N0) calculates a sum of the distance-vectors (between N0 and N1, then between N0 and N2, etc.) and makes a decision about direction to the next position by calculating an *ort* vector. In the Fig. 3, the red color marks the vector of movement in a fission process, and the blue color – the vector in a fusion process.

Red color (in Fig. 3) marks the direction of node movement in a fission process, and the blue color – the direction in a fusion process. In the simulation, all movements are discrete; in real world, they will be continual, but same principle of work still remains. Each node polls the other nodes in its *link-list*, and requests their current coordinates. After that, it calculates the movement vector and makes the decision about direction of its movement.

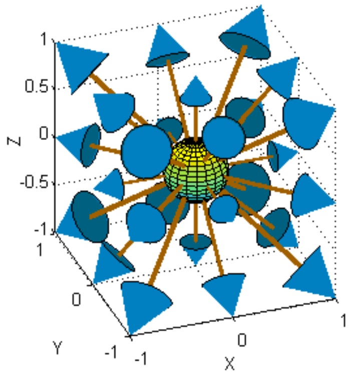

Fig. 4. Possible values of the vector of movement.

All possible values of the calculated vector for the 3D space are shown in Fig. 4. All nodes take the action at the same period determinate by an embedded timer.

B. *Fission and Fusion Simulation*

There is a simulation designed for verification of the behavior of large number of nodes in a network.

There are a number of randomly distributed nodes connected with each other, placed in the center of a 3D area. The Fig. 5 shows a screenshot which presents a distribution of network in some point of time, whether it is in the process of fission or fusion.

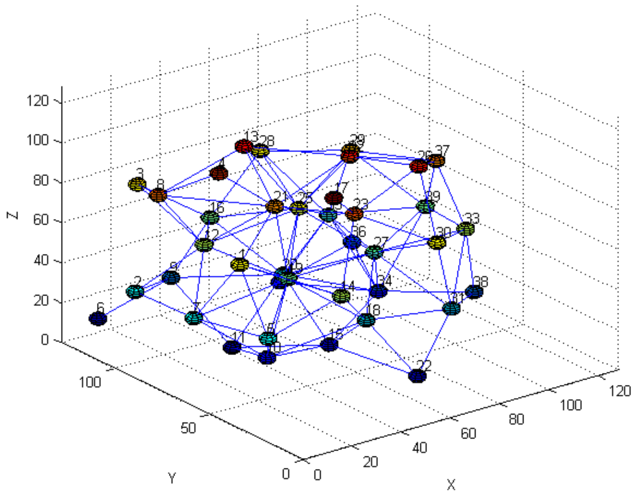

Fig. 5. Screenshot of the fission and fusion simulation.

The output video file shows the alternation of two basic processes: fission and fusion of a randomly deployed mesh network. Each of the nodes is moved at the requested distance from the other nodes in its neighborhood, but the nodes are still connected, and the consistency of the network is still preserved.

IV. IMPLEMENTATION

A. *Hardware*

Sensinode modules [4] are used because they already have a built-in location engine, but it is necessary to build the motherboard for them, so they could have more functionality.

To build module they uses RadioCrafts RC2301 chip [1], it has 8051 microcontroller core, with 32 MHz (in there is RTC, 128 kB Flash, 8 kB SRAM, 4 kB EEPROM, 19 x I/O, 8 channel 14 bit ADC, 2 x UART, SPI, I2C) and the radio chip from ex. Chipcon now TI CC2431 [5], it is the 2.4GHz SoC radio.

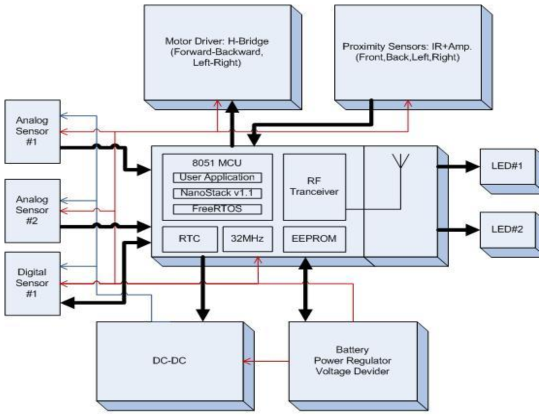

Fig. 6. S-D WSN mainboard.

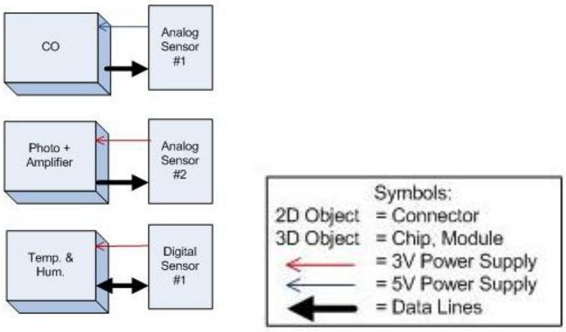

Fig. 7. Sensor interfase (on left), Symbols (right).

For motor driver we used SN754410 Quadruple Half-H driver [10]. The chip is bidirectional drive that can deliver currents up to 1A.

To detect obstacles, we used infrared sensor GP2Y0A21YK [8]. The sensor has an analog output that is approximate to a distance of the object. ADC (Analog to Digital Converter) on Sensinode is used to measure this values. For now detections is only possible for the front of the mobile unit, but from all four sides should be that sensors.

B. *Software*

NanoStack 2.0 6LoWPAN (6LoWPAN - IPv6 over Low power WPAN) is communication stack for IP based WSN [3], NanoRouter 2.0 is the gateway router software for connecting IP-based 6LoWPAN low-power wireless networks and backbone IP networks seamlessly together [3].

Sensinode's communication protocol is 6LoWPAN it is the low-power mesh technology on inexpensive radios (radio communication is the biggest power consumer in WSN).

Sensinode also has transreceiver stack – Nanostack, that can be used to communicate with other nodes, but it has to be adapted to suite the S-D WSN needs. So the Nanostack has been used, and above it we added on the *Application layer*, as shown in Fig. 8.

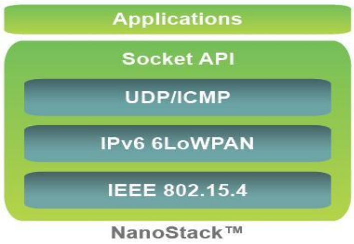

Fig. 8. Application layer on top of Nanosteck.

To process the data, gateway node is connected to the PC computer, and the PC has an application running on it. Application called NanoView is built to communicate with the gateway, but it is possible to modify it insofar that it can display nodes location in their relative positions and measurements.

Gateway node could be connect to the USB port of a PC computer or could be connected to other wireless gateway for GSM/GPRS/EDGE network [2].

C. *Sensor Modules*

The large number of sensor types can be attached to the S-D WSN device: some of them are light, CO, propane and LPN gas, PIR motion detection, metal detectors and many others. It is chosen the CO sensor 3ET1CO1500 [6], temperature and humidity sensor SHT15[7] and the light sensor (photo transistor).

CO sensor 3ET1CO1500 is an analog sensor, but it needs an Operation Amplifier (it is used AD8572 [9]) to convert the current output from sensor to voltage, so it could be measured by Sensinode ADC.

SHT15 is used to measure temperature and humidity; it is digital sensor that uses 2-wire interface. Sensor is connected directly to the Sensinode.

Photo transistor is a standard transistor with a pull-down resistor and it is connected directly to Sensinodes ADC.

D. *Computer Software*

Application that is running on PC is getting data from the gateway node. It displays the relative positions of nodes, and shows the propagation of measurements.

There are three buttons: the first refreshes displayed nodes locations and measurements (if it is needed before regular nodes reading), the second button recalls the nodes. The third button opens the *Settings* dialog.

E. *Nodes Algorithm of work*

Gateway node calls for all nodes in the range, active nodes reply by sending their address. Gateway node measures the signal strength, and according to that, it sends to computer the nodes address and the relative position. After getting their position, the gateway node asks for the measured data of each node. Both (nodes address and data) are transmitted to the computer. Computer application displays the measured data for each node in their position, relative to the gateway node. If some node is too far from the gateway node, it uses another node to send its address and the node in range measures the signal strength, and transmits that message to the gateway node. So by hopping signal all the nodes are reached.

V. Applications

In case that we have an area of a variable quantity, it can be covered by a self-deploying wireless sensor network to track the variability process. The nodes are randomly distributed in a fission process, in order to cover the area. Each node acquires a value in its point during the sampling time. It is possible to do a post-processing in order to estimate the values of the quantity in the rest of the area (it is shown on the right color plot in Fig. 9).

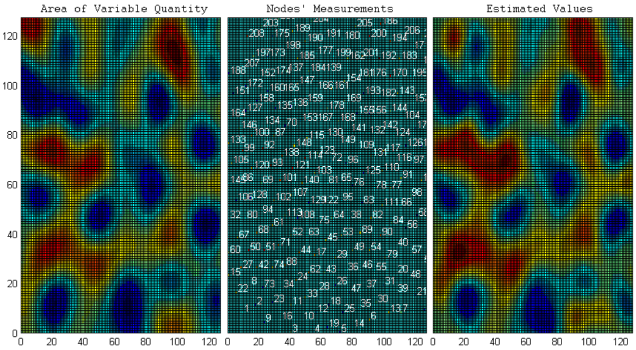

Fig. 9. Possible values of the vector of movement.

Applications for this type of WSN are numerous:

- When it is present a large number of nodes that have to be placed on predefined positions.
- If it is necessary that the time of placement has to be short (rapid deployment safes time, makes fast results and therefore safes resources).
- Measurements that can be hazardous for humans (CO e.g. distribution of exhaust gases from motor vehicles, propane e.g. search for gas leakage, radiation, metal detectors e.g. mine detection) technicians can get results from a safe distance.
- In case that the space distribution and its propagations - diffusion through the space have to be determined for certain measurements (determining the safe route for humans).
- An application of a self-deploying network on hardly accessible locations is explained in [12]. On account of that, the authors propose to use this type of network, placing the nodes onto motorized autonomous buoys with direction control system, which are supplied by a solar battery and an accumulator, so that power supply can be stable during the nights and dark days. Deploying in water (seas and oceans), the described type of network is used in order to track weather and climate changes, and to avoid tsunami disasters.

VI. Conclusion

A. Faults

Sensinode chip has only 19 I/O pins, so it is not possible to build a large system over the sensinode (without using another microcontroller and sensinode only as radio), e.g. GPRS+GPS+Sensors, it has less than 1Kb RAM available for custom application layer (8 Kb – NanoStack = < 1 Kb RAM) and less than 64 Kb of Flash after adding the Nanostack (128 K – NanoStack =< 64 Kb Flash).

Hardware Location Engine has only 3m accuracy (chip antenna, with no amplifiers) and 0.5 m resolution.

NanoStack 1.1 can relay messages up to 3 hops (NanoStack 2 up to 8 hopes) so it is not possible to deploy a larger network with this hardware.

Radio chip at 2.4GHz has a small penetration possibility (it supposedly can penetrate one wall obstacle).

It only has one gateway to the PC, Notebook or PDA.

B. Advantages (current design)

Hardware Location Engine has the location range 64 x 64 m (with a small chip antenna) and the estimation time of positioning less than 40 us.

Sensinode module has 8 channel 14-bit ADC, the SPI, I2C and UART interface so it is possible to connect various types of sensors. The major advantage is that module comes with complete stack for IP and Router, this means that we only have to write an application layer.

Module also has a very low power consumption (27 mA, in PowerDown mode only 0.6 uA) and low power supply (2.0 - 3.6V). Data transfer rate is 250Kb/s.

C. Upgrade

Separate antenna and good amplifier with the radio module would give much wider location range and much better resolution (CC1110 radio chip with 868/915 MHz would afford the longer range).

Future plans are to build a database which will store the data, and application for graphic analyzation and presentation the sensor data.

References

[1] RC2300/01/02/04 ZigBee™- Ready RF Transceiver Modules http://www.radiocrafts.com/uploads/rc230x_data_sheet_1_2.pdf

[2] RC-IESM-Series, Wireless M-Bus and ZigBee Expansion cards for Wavecom Fastrack Supreme GSM/GPRS/EDGE modems http://www.radiocrafts.com/uploads/rc_iesm_shortform_datasheet_1_0.pdf

[3] NanoStack 2.0, NanoRouter 2.0, NodeView Pro http://www.sensinode.com/EN/products/software.html

[4] Sensinode, 6LoWPAN DevKit IP-based Wireless Sensor Networking http://www.sensinode.com/EN/products/evaluation-kits.html

[5] CC2431 low-power RF SOC http://www.ti.com/corp/docs/landing/cc2431/index.htm

[6] 3 Electrode T1 Series Carbon Monoxide Sensor – 3ET1CO1500 http://www.kwjengineering.com/products/carbon-monoxide/sensors-for-carbon-monoxide/3et1co1500

[7] SHT15 – Digital Humidity Sensor http://www.sensirion.com/en/01_humidity_sensors/03_humidity_sensor_sht15.htm

[8] GP2Y0A21YK wide-angle distance measuring sensor. http://www.sharpsma.com/Page.aspx/americas/en/part/GP2Y0A21YK/

[9] AD8572 Zero-Drift, Single-Supply, Rail-to-RailInput/Output Operational Amplifiers, http://www.analog.com/static/imported-files/Data_Sheets/AD8571_8572_8574.pdf

[10] SN754410 is a quadruple high-current half-H driver http://focus.ti.com/lit/ds/symlink/sn754410.pdf

[11] X. Xu and S. Sahni, "Approximation Algorithms For Wireless Sensor Deployment", Department of Computer and Information Science and Engineering, University of Florida, Gainesville, April 21, 2006, published

[12] G. Werner-Allen, K. Lorincz, and M. Welsh, Harvard University, Omar M. and J. Johnson, University of New Hampshire, M. Ruiz and J. Lees, University of North Carolina, "Sensor-Network Applications - Deploying a Wireless Sensor Network on an Active Volcano", March - April 2006, published