Cellular Automata for Topology Control in Wireless Sensor Networks

Stavros Athanassopoulos and Christos Kaklamanis and Panagiota Katsikouli and Evi Papaioannou
Computer Technology Institute & Press "Diophantus"&
Department of Computer Engineering and Informatics
University of Patras, GR26504 Rion, Greece
E–mail: {athanaso, kakl, katsikouli, papaioan}@ceid.upatras.gr

***Abstract*— We use cellular automata for simulating topology control algorithms in Wireless Sensor Networks (WSNs). A cellular automaton is a decentralized computing model providing an excellent platform for performing complex computations using only local information. WSNs are composed of a large number of distributed sensor nodes operating on batteries; the objective of the topology control problem in WSNs is to select an appropriate subset of nodes able to monitor a region at a minimum energy consumption cost thus extending the network lifetime.**

We have used cellular automata to model a randomized WSN topology control algorithm and have experimentally evaluated its performance.

Keywords: cellular automata, Wireless Sensor Networks, topology control algorithms, simulation, randomization.

I. Introduction

A. Cellular Automata

Cellular automata (CA) are an idealization of a physical system where space an time are discrete and the physical quantities take only a finite set of values. They consist of a regular grid of cells, each in one of a finite number of states (like, for instance, On/Off). The grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood (usually including the cell itself) is defined relative to the specified cell. For example, the neighborhood of a cell might be defined as the set of cells at distance 2 or less from the cell. An initial state ($t = 0$) is selected by assigning a state to each cell. A new generation is created (increasing t by 1), according to some fixed rule (generally, a mathematical function) that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. For example, the rule might be that the cell is "On" in the next generation if exactly two of the cells in the neighborhood are "On" in the current generation, otherwise the cell is "Off" in the next generation. Typically, the rule for updating the state of cells is the same for each cell and does not change over time, and is applied to the whole grid simultaneously, though exceptions are known.

Formally, a CA is a 4-tuple (C, Σ, N, f), where C denotes a d-dimensional array of cells or lattice (cells are indexed by vectors from Z^N), Σ denotes the alphabet giving the possible states each cell may take, N denotes the neighborhood (i.e. $N \subset Z^d$) and f denotes the transition function of type $\Sigma^N \to \Sigma$. The state of all cells in time is called configuration. Significant configuration is the starting configuration, since it has to be provided with the CA.

CA is a discrete computational model, which is capable to provide the same computational power as Turing Machine, therefore it is Turing Complete. CA were firstly used by Jon von Neumann in late 1940s when he was trying to describe self-reproducing automaton. He succeeded by introducing two dimensional Von Neumann's cellular automata with rules and starting configuration such that after certain amount of time steps there were two copies of the pattern from starting configuration and so on. CA have received extensive academic study into their fundamental characteristics and capabilities and been applied successfully to the modelling of natural phenomena. In this respect, two notable developments can be credited to Conway and Wolfram. In the 1970, the mathematician John Conway proposed his now famous Game of Life [8] which received widespread interest among researchers. Conway's CA involves a 2-dimensional infinite grid of cells where each cell has two possible states, dead or alive, and simulates the evolution of a population using 4 basic transition rules. A neighborhood consists of the middle cell and the eight cells surround it (Moore neighborhood) [4], [8], [9] unlike the von Neumann neighborhood [4], [9] that contains a cell together with the four cells in the four directions attached to it. Later on, in 1980s, Stephen Wolfram [16] defined four classes of cellular automata depending on complexity and predictability of their behaviour; he has also studied in much detail a family of simple one-dimensional CA rules (known as Wolfram rules [15]) showing that even these simplest rules are capable of emulating complex behavior.

The simple structure of cellular automata has attracted researchers from various disciplines. Cellular automata have been studied in the context of several scientific areas like computability theory, mathematics, physics, theoretical biology and microstructure modelling. CA have been subjected to rigorous mathematical and physical analysis and their application has been proposed in different branches of science - both physical and social. In particular, CA have been suggested as appropriate models in many application contexts, like public key cryptography [1], channel assignment in mobile networks [2], pattern recognition [7], games like the Firing Squad [15] etc. Furthermore, CA have been used in medical applications regarding the growth of tumors [3], the implementation of the immune system [14] and the treatment of HIV [13]. Other applications of CA include the simulation of natural phenomena [10], urban growth [6], behaviour of a population in a certain situation [11] etc. The reason behind the popularity of CA can be traced to their simplicity, and to the enormous potential they hold in modeling complex systems, in spite of their simplicity [7]. Cellular automata can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components (cells). The communication between constituent cells is limited to local interaction. Each individual cell is in a specific state which changes over time depending on the states of its local neighbors. The overall structure can be viewed as a parallel processing device. However, this simple structure when iterated several times produces complex patterns displaying the potential to simulate different sophisticated natural phenomena.

B. Topology control in Wireless Sensor Networks

A WSN is a special kind of network composed of a large number of autonomous sensor nodes geographically scattered on a surface with the ability to monitor an area which is inside their range and collect data about physical and environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants. A source collects this data and can be located anywhere in the network. WSNs were initially used by the army for tactical surveillance without the need of human presence; however, WSNs have been used in a wide range of applications, such as environmental monitoring, industrial process monitoring and control, machine health monitoring, etc.

Important characteristics of WSNs include low computational power, low computational speed, small bandwidth, limited memory, limited energy, high failure tolerance, no demand for human or artificial supervision.

The most important performance aspect in WSNs is the need to be energy efficient as sensor nodes have a finite energy reserve offered by a battery. Topology control is a technique used mainly in WSNs to reduce the initial topology of the network in order to save energy, cut down interference and extends the lifetime of the network [12]. The objective of the topology control problem in WSNs is to discover a minimum configuration of nodes capable of monitoring a region equivalent to that one monitored for all nodes with an aim to eventually ensure a longer lifetime for the network. This is possible because in WSNs there is a lot redundancy (that is, two or more nodes monitoring the same region) due to a random node deployment. Due to this, redundancy problems such as energy waste, packet collisions and congestion may arise. Therefore, the network topology must be controlled to avoid these negative effects.

Efficient topology control techniques in WSN are very critical and essential: sensors operate on limited energy (i.e., batteries). Managing this scarce resource efficiently by controlling the network topology directly influences (i.e., extends) the network lifetime. Moreover, despite the existence of a rich literature of theoretical approaches to the topology control problem in WSN, heuristics are important since they usually capture practical aspects of problems rather than their fundamental limitations and may lead to efficient techniques in practice (which are usually adopted by industry and used in practice since they impose a lower cost offering on average satisfactory performance).

We have used cellular automata to model a variation of the WSN topology control algorithm proposed in [17] using randomization and have experimentally evaluated its performance. In [17] experimental tests presented do not use cellular automata and are implemented using PARSEC[1]. In [5], another variation of the topology control algorithm of [17] has been simulated using cellular automata; however, in [5], they assume that idle nodes become active in a deterministic way. In our work, idle nodes become active according to the outcome of a weak random source; moreover, we assume homogenous networks, i.e., there are no structural differences between the nodes and all sensors have the same hardware and software features.

In Section II, we present the topology algorithm and the cellular automaton we used for its simulation; we present our experimental framework and our simulation results in Section III.

II. Topology control algorithm and cellular automaton description

The main idea of the WSN topology control algorithm (as sketched in [17]) is that network nodes are activated only if there are few neighboring active nodes; otherwise they remain idle saving their energy. In particular, initially, sensors are placed on the grid according to a sensors deployment scheme and are all active. Every active node, in every step, counts its active neighbors: if there are at least two active neighbors, the node becomes idle; otherwise, the node remains active. Every idle node, in every step, uses a weak random source (i.e., it casts a five-sided "die") in order to decide whether it will remain idle or turn to active. Initially, each node has 0.8 units of energy; energy consumption is assumed to be 0.0165 units/step for active nodes and 0.00006 units/step for idle nodes [17], [5]. A node turns off when it runs out of energy. The algorithm terminates when there is no alive (active or idle) node in the network.

For our cellular automaton, we have assumed Moore neighborhoods [4], [8], [9]: the neighborhood of each node consists of its 8 surrounding nodes. We developed an Outer-Totalistic cellular automaton[2] for our simulation. Transition rules of Outer-Totalistic CA are based on the sum of the states of the neighbors. For instance, the state of cell u at time $t+1$ is calculated based on the sum of its state together with the states of its neighboring cells at time t [9]. In order for "sums of states" to be defined, we have represented states by integers. In particular, we have assigned values 1 to active cells, 0 to

[1] Parsec is a C-based simulation language, developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models (http://pcl.cs.ucla.edu/projects/parsec/).

[2] A class of CA to which Life belongs.

idle cells and 2 to dead or empty cells; in this way, we count active cells by increasing a counter. Regarding the transition of our CA, every active cell, at every time unit, counts its active neighbors. An active node with at least two active neighbors turns idle. An active node with one or no active neighbors remains active. An active neighbor with no battery turns off. An idle neighbor turns active at a random moment and performs again these rules. Every idle node, in every step uses a weak random source in order to decide whether it will remain idle or turn to active. For the implementation of the CA rules, we have used boundary conditions, i.e., conditions applied to cells at the borders of the grid [4]. In particular, we added a hidden extra raw of cells around the main grid: these extra cells help in the formation of the neighborhoods but are not used in calculations.

III. Implementation details and simulation results

For the experimental evaluation of our algorithm, we developed a Java application that performs simulation and visualization. Fig. 1 shows the GUI of our WSN topology control application. There are two parts: the grid and the user menu. The two-dimensional grid represents the WSN. Every cell is a possible place for a sensor node. The user menu consists of drop-down menus for sensor deployment scheme and network size options as well as start and pause options.

The application receives as input the network size and the sensor deployment scheme selection; it computes and visualizes the evolution of the WSN when our topology control algorithm is applied to it for its entire lifetime; it also returns number of active and idle nodes for the WSN lifetime.

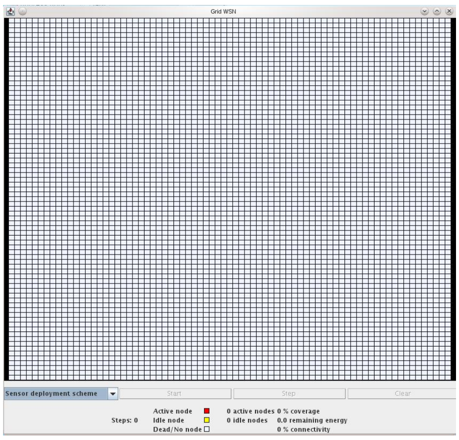
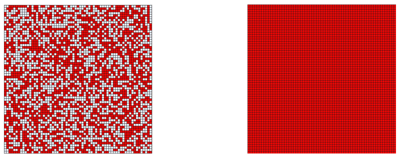


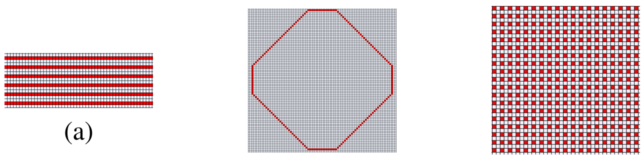
Fig. 1. The GUI of our application for a WSN of size 75 × 75.

Currently, in the beginning of the application, the user is requested to select an integer n in [5,300] in order to produce a grid WSN of size $n \times n$. We have implemented 5 sensor deployments schemes: random and full, rows and circles deployment and sparse deployment. In random deployment 2(a), active nodes are placed randomly on the grid WSN without necessarily covering every possible cell. Full sensor deployment implies that every cell of the grid contains an active node 2(b). Random and full deployment offer high connectivity among nodes. In rows deployment nodes are placed one next to the other, as shown in Fig.3(a), each sensor forwards information to its neighbor so that the final sensor receives the data. In circular deployment, nodes form a circle, as shown in Fig.3(b); such a deployment could be useful to model the protection of an area from surrounding dangers. Finally, sparse deployment, as shown in Fig.3(c), forces all sensors to remain active until their energy is exhausted. Cells are characterized by their available energy amounts (battery) and their state at every time step.



(a) (b)

Fig. 2. Initialization of the grid WSN using random (a) and full (b) deployment of sensors. Red cells are active nodes, yellow cells are idle nodes, white cells are dead or no nodes.



(a) (b) (c)

Fig. 3. Initialization of the grid WSN using rows (a), circular (b) and sparse (c) deployment of sensors. Red cells are active nodes, yellow cells are idle nodes, white cells are dead or no nodes.

The application was developed in Sun Java 1.6.0.u26 and executed on a OpenSUSE 11.4 (Linux Distribution) system with AMD Athlon II x4 640 Processor at 3GHz and 3.6 GHz memory. Figures were plotted using the Open Source tool Octave (http://www.gnu.org/software/octave/).

Simulation results for our algorithm in a WSN of size 300×300 for full and random deployment are presented in Fig. 4, 5, 6, compared to the case when all sensors are active until their energy is exhausted (i.e., no topology control algorithm is applied).

In our simulation, we have used a cellular automaton to model a WSN topology control algorithm; good news is that our results are inline (and actually slightly better) with those presented in [5]. We observe that application of our algorithm - which uses randomness for the activation of idle nodes - prolongs network lifetime compared to what happens when no algorithm is used (i.e., when all sensors remain active until their energy is exhausted): the total energy of the WSN falls relatively slow thus extending network lifetime. In particular (Fig. 4), when full sensor deployment is used, network lifetime is extended 4.5 times more that when no topology control algorithm is used; in [5], they assume full sensor deployment and they obtain an extension in network lifetime 4 times more that when no topology control algorithm

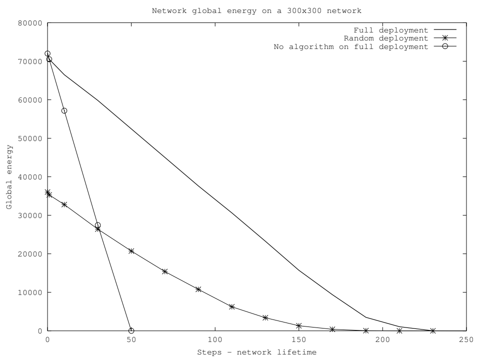


Fig. 4. Simulation results for our algorithm in a WSN of size 300 × 300: total network energy.

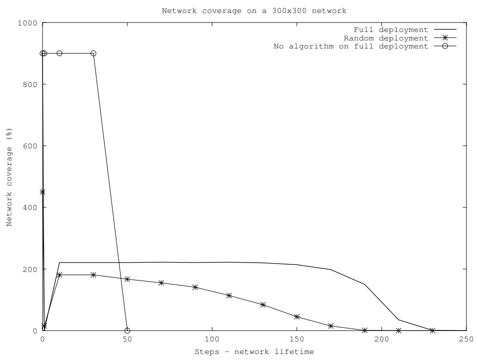


Fig. 5. Simulation results for our algorithm in a WSN of size 300 × 300: coverage.

is used. When random sensor deployment is used, network lifetime is extended 4 times more that when no topology control algorithm is used.

Full deployment of sensors (i.e., sensors are placed one in every network node) improves coverage compared to random sensor deployment or to the no topology control case; in particular (Fig. 5), our algorithm obtains a coverage of more than 80% for most of the network lifetime. Regarding connectivity (Fig. 6), as the WSN energy decreases, there is high clustering in the WSN (active sensors create disjoint connected groups) resulting in low connectivity. Since WSN energy levels are not too low, we could fix this inefficiency by letting the algorithm check a larger neighborhood before deciding which nodes to keep idle or active. It is worth investigating whether adopting an alternative neighborhood pattern for our cellular automaton (like for example a denser Margolus neighborhood) would help in increasing connectivity.

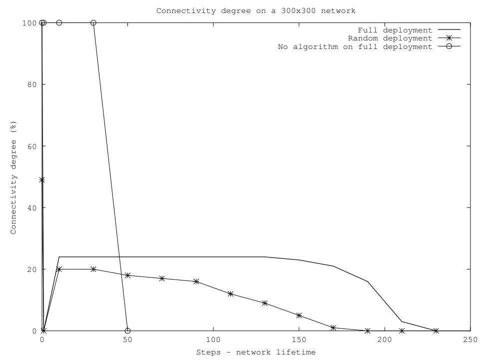


Fig. 6. Simulation results for our algorithm in a WSN of size 300 × 300: connectivity.

ACKNOWLEDGMENT

This work was partially supported by the EU project ICT-2010-258307 EULER and by the EU and the Greek Ministry of Education, Lifelong Learning and Religious Affairs project "Digital School" (296441).

REFERENCES

[1] B. Applebaum, Y. Ishai and E. Kushilevitz. Cryptography by Cellular Automata or How Fast Can Complexity Emerge in Nature? In Proc. of the 1st Symposium on Innovations in Computer Science (ICS 10), Tsinghua University Press, pp. 1-19, 2010.

[2] H. Beigy, M. R. Meybodi. A self-Organizing Channel Assignment Algorithm:A Cellular Learning Automata Approach. In Proc. of the 4th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 03), LNCS 2690, Springer, pp. 119-126, 2003.

[3] A. Boondirek, W. Triampo and N. Nuttavut. A review of Cellular Automata Models of Tumor Growth. *International Mathematical Forum*, vol. 5, no. 61, pp. 3023 - 3029, 2010.

[4] B. Chopard, M. Droz. Cellular Automata Modeling of Physical Systems. Cambridge University Press, 1st Edition, 1998.

[5] R. O. Cunha, A. P. Silva, A. A. F. Loureiro and L. B. Ruiz. Simulating Large Wireless Sensor Networks Using Cellular Automata. In Proc. of the 38th Annual Simulation Symposium (ANSS 05), IEEE Computer Society, 2005.

[6] H. Demirel and M. Cetin. Modelling urban dynamics via cellular automata. In Proc. of the 7th AGILE Conference on Geographic Information Science, pp. 313-323, 2010.

[7] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. Pal Chaudhuri. A survey on cellular automata. Technical report. Centre for High Performance Computing, Dresden University of Technology, December 2003.

[8] M. Gardner. The Fantastic Combinations of John Conway's New Solitaire Game 'Life'. *Scientific American*, Vol. 223, pp. 120-123, 1970.

[9] A. Ilachinski. Cellular Automata: A Discrete Universe. World Scientific Publishing Co. Pte. Ltd, 57 Shelton Str, Covent Garden, London, UK, 1st Edition, 2001.

[10] G. Iovine, S. Di Gregorio and V. Lupiano. Debris-flow susceptibility assessment through cellular automaton modeling: an example from 15-16 December 1999 disaster at Cervinara and San Martino Valle Caudina (Campania, Southern Italy). *Natural Hazards and Earth System Sciences*, Vol. 3, pp. 457468, European Geosciences Union, 2003.

[11] Y. Lizhong, F. Weifeng, L. Jian, H. Rui and F. Weicheng. Cellular automaton pedestrian movement model considering human behavior. *Chinese Science Bulletin*, Vol. 48, No. 16, pp. 1695-1699, 2003.

[12] P. Santi. Topology Control in Wireless Ad-hoc Networks. *ACM Computing Surveys Surveys*, Vol. 37, Issue 2, pp. 164-194, 2005.

[13] P. Sloot, F. Chen and C. Boucher. Cellular automaton model of drug therapy for HIV infection. In Proc. of the 5th International Conference on Cellular Automata for Research and Industry (ACRI 02), LNCS 2493, Springer, pp. 282293, 2002.

[14] T. Tome and J. R. D De Felicio. Probabilistic cellular automata describing a biological immune system. *Physical review*, Vol. 53, Issue 4, pp. 39763981, 1996.

[15] S. Wolfram. Theory and Applications of Cellular Automata. World Scientific, Singapore, 1986. ISBN 9971-50-124-4 pbk.

[16] S. Wolfram. A New Kind of Science. Champaign, IL: Wolfram Media, pp. 29-30, 52, 59, 317, and p. 871, 2002.

[17] F. Ye, G. Zhong, S. Lu and L. Zhang. Energy efficient robust sensing coverage in Large Sensor Networks. UCLA Technical Report, 2002. URL: http://lion.cs.uiuc.edu/coverage-tech-report.ps.