GENSEN: A Topology Generator For Real Wireless Sensor Networks Deployment

Tiago Camilo[1], Jorge Sá Silva[1], André Rodrigues[1], Fernando Boavida[1]

[1] Department of Informatics Engineering, University of Coimbra
Polo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
{tandre, sasilva, arod, boavida}@dei.uc.pt

Abstract. Network Simulators are important tools in network research. As the selected topology often influences the outcome of the simulation, realistic topologies are required to produce realistic simulation results. The topology generator presented in this document, GenSeN, was created based on the authors' knowledge from several experiences. GenSeN is a tool capable of generating realistic topologies of wireless sensor networks and, additionally, auto-configuring important characteristics of sensor nodes, such as energy parameters. The tool was validated by comparison with real deployment strategies and experiences.

Keywords: Wireless Sensor Networks, Simulators, Sensor Node Deployments.

1 Introduction

Due to technology advances in telecommunications, microprocessors and monitoring, it is now possible to design networks with special features, such as Wireless Sensor Networks (WSNs) [1]. Such networks have specific requirements such as reduced energy availability, low memory and reduced processing power. A WSN consists of a number of sensors (e.g. from ~10 up to ~10000) spread across a geographical area. Each sensor is equipped with a wireless communication system, and some level of intelligence for signal processing and networking of the data.

Although they can be considered ad hoc networks, WSNs are in fact quite distinct from these networks in the deployment phase. In typical ad hoc networks, devices are mobile and their location is a random factor, since users of such equipment normally cannot predict the place and time where the network will be stable. On the other hand, in WSNs the deployment phase is critical and may require careful planning, due to the singular characteristics of sensor nodes. As the authors demonstrated [2], the correct distribution of sensor devices over the target monitoring area affects the entire WSN deployment, from the choice of the correct sensor nodes, to the correct network protocol, and the architecture/topology to use. When a WSN solution is designed, it is important to define the main evaluation criteria that, in the end, will be used to validate the obtained results. Lifetime, latency, fault-tolerance, scalability and precision are some of the parameters used to evaluate WSN solutions.

In optimal conditions, where radio interference does not exist, the terrain is plane and no obstacles are present, the most effective deployment strategy would be the grid strategy, where all devices are in range and evenly placed to cover the whole monitoring area. However, due to their vast applicability, WSNs are commonly deployed in unusual locations, where human accessibility is limited (e.g. inside a volcano). In such environments it is necessary to place sensor nodes using different strategies (e.g. dropping sensor nodes from a plane). Therefore, it becomes crucial to develop the necessary tools to study such variables in a simulation environment, before the tests in the final environment begin. In [13] the authors identified some research issues / directions that influence the WSN MAC layer development. One of them was the need to improve the simulation tools with better representations of the reality (namely better radio models that account for terrain, antenna location, foliage types, etc). This paper, based in data from [2,8], tries to address some of these problems by developing a new WSN topology generator for Network Simulator 2 (NS-2).

The remainder of this paper is organized as follows: Section 2 presents the related work regarding topology generators. Special focus is given to the topo_gen, which is the only specific tool that covers WSNs. Section 3 identifies the main problems regarding the deployment of WSN and presents six different deployment strategies. In Section 4 the GenSeN is introduced. Special attention is given to the input and to the output parameters of the generator. Results taken from the topology generator are discussed in Section 5. Finally, conclusions and future work are addressed in the last section.

2 Related Work

The development of topology generators that emulate real environment features is a problem that attracted and still attracts the attention of the scientific community. Nowadays there are a number of competing approaches to the construction of random network topologies, for wired and wireless environments.

One of the most popular generators available is BRITE [3]. It is a flexible tool that supports flat router and hierarchical topologies, allowing the configuration of several important parameters such as bandwidth and delay.

GT-ITM [4], another topology generator, focuses on reproducing the hierarchical structure of the topology of the Internet.

Finally, the Inet [5] is a network generator aiming to reproduce the connectivity properties of Internet topologies, assigning node degrees from a power-law distribution.

However, the referred approaches are specially designed to build well known Internet network topologies, which present significant differences when compared to WSNs.

The Topo_gen [11] is a topology generator designed by the ISI Laboratory for Embedded Networked Sensor Experimentation, which intends to be a tool to generate random sensor node locations. Although it was originally designed to be used in directed diffusion experiments, it is an adaptable tool that can be easily ported to support other protocols. Topo_gen has some configurable parameters such as map dimensions, source and sink count. It allows the creation of topology files for NS-2 and EmSim [12]. Nonetheless, this topology generator does not take in account real sensor network placements, since it only allows random or cluster node distribution. Characteristics such as sensor node deployment strategies are not covered.

3 Wireless Sensor Network Deployment

WSNs differ from typical ad hoc networks by requiring a deployment phase, in contrast with ad hoc networks, which are known to group spontaneously and move in a random way. Sensor nodes are normally placed in special environments without guarantee of position. This is why the deployment phase in a WSN project is extremely important to the final experiment output. Due to its characteristics, the WSN can be deployed in environments where the accessibility by humans is difficult, and where ambient conditions can significantly vary.

The minimal number of sensor nodes required to monitor a specific area (A), is provided by Equation 1, where r represents the sensing range of each sensor node [6].

$$NS = \frac{2A.\pi}{r^2\sqrt{27}} \tag{1}$$

However, this approach considers that all nodes have the same monitoring capabilities, which means that it cannot be applied to WSN with different types of sensor nodes. Moreover, it does not take into account the existence of obstacles, such as trees or walls.

As mentioned before, in order to optimize sensor node placement, sensor nodes must be deployed as a grid, where all devices are meticulously and evenly spaced according to their monitoring/transmitting range. Such method minimizes the number of nodes needed to monitor a specific area, with full phenomenon coverage. However, such scenario can only be applied when ideal radio environment characteristics are present (i.e. no radio interference exists, the terrain is flat and there is no vegetation). Moreover, sensor nodes deployment is often made in inhospitable locations, where it becomes impossible to deploy a uniform distribution. In places such as the ocean bed, it is not feasible to deploy sensor nodes in a grid arrangement. On the other hand, applications such as monitoring a cyclone require a fast deployment phase, since this kind of phenomenon is not predictable, and it is necessary to distribute sensor nodes as quick as possible to maximize the coverage area, connectivity, etc.

For this reason, it is necessary to consider different deployment strategies, which could be used in inhospitable areas or could be more suitable to different sensor network applications. Therefore the authors suggested six different deployment strategies in a previous piece of work [2], each one listed below:

- **Grid**: in this strategy it is important to create a network of sensors similar to the one illustrated in Fig. 1, where each sensor device is evenly separated from neighboring devices by r, which is the communication/monitoring range of each sensor. An operator is responsible to place each sensor facing up (antenna point up). A ribbon-metric is used so that the sensor location is determined as exactly as possible;

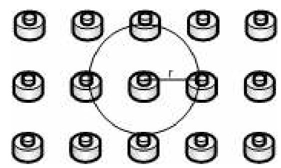

Fig. 1 Sensor nodes placed as a grid

- **One-by-One**: this strategy consists in deploying each sensor individually, but without using metrics tools. The operator is responsible to throw each node using only his knowledge regarding the average distance between nodes, not relying on the node position;
- **Two-by-two**: the only difference between this strategy and the previous one, is the fact that the operator, in this case, throws sensor nodes in pairs;
- **Three-by-three**: the technique is similar to that in the one-by-one and two-by-two strategies, but in this case the operator drops the sensor nodes in groups of three elements;
- **Cliff**: in this strategy the operator drops the nodes from a higher point, more precisely from a 10 meters crag. Such strategy intends to simulate a WSN experiment were the phenomenon is located far bellow the operator, or even to simulate the deployment from inside a helicopter;
- **Propellant**: in this final strategy the sensors are spread in the area to monitor through the help of a propellant. All sensors are spread at the same time. The propellant is calibrated to send the sensors nodes to the middle of the monitored area.

The different deployment strategies were compared using a real WSN implementation [8]. In a 60 m2 monitoring area (6 per 10 meters), a set of Embedded Sensor Board (ESB) sensor devices from the ScatterWeb [9] platform were used. The environment was strategically chosen: plain, dry and with no natural or human made obstacles. Each sensor was placed at ground level. For each strategy, the deployment time, cost and network connectivity were analyzed. By dividing the rectangle area in 15 squares the authors found the average node location per square, on each deployment strategy; the results can be found in [2]. From this study it was possible to conclude that the best results were obtained by the grid strategy, since it leads to an optimal node distribution, (each of the 15 regions was covered by one node). Strategies such as cliff and propellant, which have reduced deployment time, tend to concentrate the nodes in the center of the scenario, decreasing the area covered by the sensors.

Another important difference in WSN deployment is the fact that it is not possible to guaranty the correct node (antenna) orientation, contrary to normal ad hoc networks behaviour. When a node is spread (e.g. by a propellant), depending on the device (the ESB permits six different antenna orientations), it can be facing different positions: antenna up, antenna down, etc. As presented in [2], such characteristic can be crucial for WSNs, since a bad sensor node position can decrease the radio range by 30%.

4 GenSeN: A Generator for Sensor Networks

The need for a realistic WSN topology generator has long been recognized by sensor network researchers. Such tool, associated to a network simulator, is the first instrument to understand the behavior of new protocol prototypes. The existing approaches, as described in Section 2, are not adequate to the WSN characteristics (ad hoc network topology generators), or do not contemplate realistic node distribution (TopoGen).

The GenSeN is a topology generator built in C++ that is specifically designed to work with NS-2 [7]. It is based on real WSN deployments performed by the authors, and described in Section 3. This generator has the capability to simulate the behaviour of the presented deployment strategies: grid, one-by-one, two-by-two, three-by-three, cliff, and propellant. It presents several input parameters which are used to characterize each sensor node (Fig. 2).

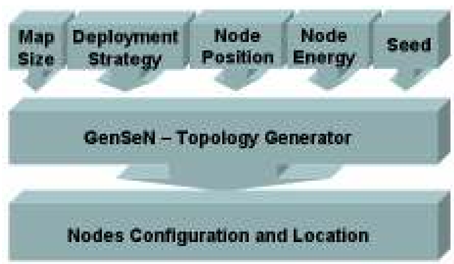

Fig. 2 GenSeN – Topology Generator Architecture

As output, GenSeN provides a tcl format file containing the configuration of each sensor and also its position in the monitoring area.

The following sub-chapters provide a detailed explanation on the input and output parameters provided by GenSeN.

4.1 GenSeN Input

GenSeN provides variable input parameters, each one used in the characterization of each sensor device. As illustrated in Fig. 2 the first parameter to be introduced is the monitoring area dimension. Unfortunately NS-2 does not yet provide a tridimensional scenario. For this reason only x and y parameters are used.

In the next step it is necessary to choose which will be the deployment strategy to use in the sensor distribution. The user has the following options:

```
1 - Grid (You need to set the distance)
2 - Random (DEFAULT)
3 - One-by-one
4 - Two-by-two
5 - Three-by-three
```

```
6 - Propellant
7 - Cliff
```

The user has the possibility to choose any of the deployment strategies presented in the previous section, plus the random node distribution, which is in fact the default option. This latter option will randomly distribute the sensors throughout the defined monitoring area. Such option should be used in case the user does not know which deployment strategy will be used in the final WSN implementation. In the grid alternative the user must set the minimal distance between the nodes. Such distance should be set to the smallest of two distances: the radio range and the sensing scope. The node locations are determined by the probabilistic rules learnt from [2].

Regarding the node position, GenSeN enables the user to configure the number of possible node orientations (regarding the antenna). The user should select how many possible positions the sensor node device will have. As an example, the device used in [2], the ESB, only supports four possible orientation-stages, contrary to the Mica family (Mica2, Micaz), which presents only two. As default, only one position is considered:

```
1 - One Position (DEFAULT)
2 - Two Positions
3 - Three Positions
4 - Four Positions
```

GenSeN will randomly choose the node position, which then will affect the transmitting and receiving energy for each node. Due to restrictions in NS-2, it is not possible to specify different radio propagation conditions in the same simulation. Therefore, it was necessary to emulate such capability. This was achieved by modifying the levels of transmitted energy in each node, although all the nodes have the same transmitting range.

In terms of energy configuration, GenSeN allows to set the following parameters: initial energy, idle energy, transmitting energy and receiving energy. Moreover, it supports the configuration of four different levels of energy, meaning GenSeN can generate different initial energy levels per sensor node. To allow such behavior, the user needs to choose one of the following options:

```
1 - All nodes with same energy (DEFAULT)
2 - Two energy levels
3 - Three energy levels
4 - Four energy levels
```

For each node, GenSeN will randomly choose the initial energy based on the option (initial energy) provided by the user.

In each iteration, GenSeN generates different results when compared to previous iterations, since its random factors are associated to a variable seed. However, the generator enables the user to configure its own seed, allowing debugging operations. Finally, it is also possible to specify the final *tcl* output file.

4.1 GenSeN Output

GenSeN is NS-2 based, meaning it was created to produce a WSN topology which can be used in a simulation experience in this simulator. As such, GenSeN outputs two tcl script files. One of the files saves the information regarding the nodes location (*x* and *y*), which is created as result of the chosen deployment strategy. The second file outputs the node configurations, namely the energy parameters of each individual node.

Finally, GenSeN calculates the estimated time required to perform the deployment of the entire sensor network which, once again, depends on the deployment strategy chosen by the user. These values are based on the ones achieved by the authors in [8].

5 Results

In order to study the behaviour of GenSeN, this section presents the results produced by this topology generator, and briefly validates them with the results achieved by the authors in [2].

The main goal of this study is to compare the node distribution for the different deployment strategies. Using a monitoring area of 6000 m2 (100 m per 60 m), a total of 32 sensor nodes were spread using seven deployment strategies: grid, random, one-by-one, two-by-two, three-by-three, cliff, and propellant, as described in Section 3. The environment area was virtually cut into 16 equal squares, each one with 375 m2. The number of nodes distributed on each square was registered and it is illustrated in figures 3-9. Since there are 32 sensor nodes, it would be expected that the best solution were to deploy 2 devices per region. Such result was achieved only by the grid strategy (Fig. 3). This solution presents the best results in terms of connectivity and sensing coverage.

Fig. 4 presents the results produced by the random distribution of nodes. This technique is the most used by the scientific community when simulating new protocols. However, the results are not very encouraging, since there exist enormous differences in terms of area coverage per device, and it also does not really reflect a real deployment strategy.

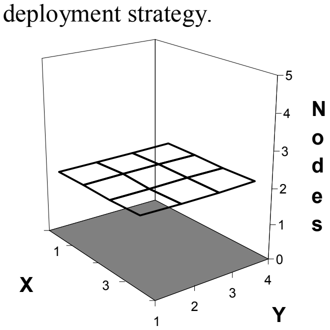

Fig. 3. Nodes per region for the Grid Deployment Strategy

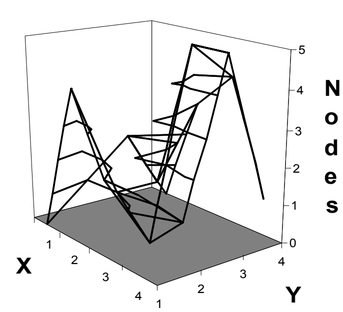

Fig. 4. Nodes per region for the Random deployment strategy

On the other hand, the solution presented in **Error! Reference source not found.**, shows a better placement when compared to that of Figure 4. In Figure 5 the maximum numbers of nodes per square is 3, and all the squares are covered by at least one device.

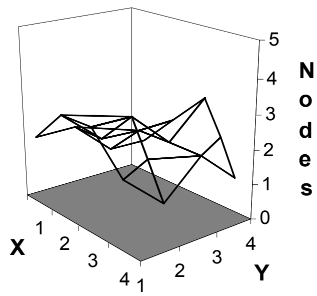

Fig. 5. Nodes per region for the One-by-one deployment strategy

Fig. 6 and Fig. 7 present the two-by-two and three-by-three strategy deployments, respectively. These strategies lead to poorer results when compared to the one-by-one strategy. In the case of the three-by-three strategy, one of the square areas ends up with no devices at all. On the other hand, a single region ends up with 5 devices.

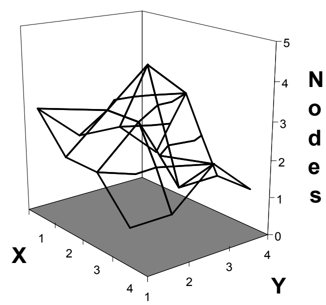

Fig. 6. Nodes per region for the Two-by-two deployment strategy

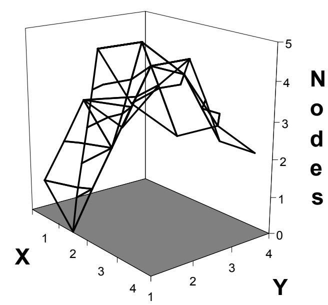

Fig. 7. Nodes per region for the Three-by-three deployment strategy

Finally, Fig. 8 and Fig. 9 show the worst sensor distribution results, mainly due to the characteristics of the used strategies. It is notorious that there is an excessive node density in the centre of the environment area, contrary to the edges, where several regions without sensor nodes exist.

Looking at the presented results in terms of sensing coverage, there is a huge difference between deployment strategies. This is why it is important to consider realistic models when developing new network protocols. All the results are coherent with the ones achieved in [2]. The strategies that require human intervention in the monitoring area achieved the best results, as opposed to the last two approaches where the nodes appear disorganized.

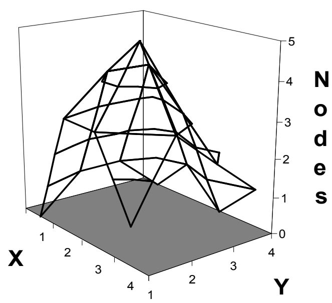

Fig. 8. Nodes per region for the Cliff deployment strategy

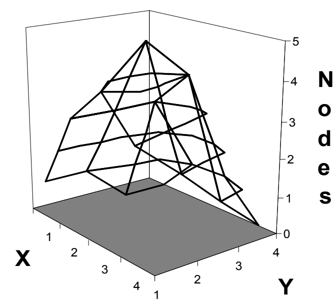

Fig. 9. Nodes per region for the Propellant deployment strategy

5 Conclusions

In a WSN project, it is important from an early stage to define the deployment strategy to use. Decisions such as which architecture to use or what kind of nodes to deploy, have to be in consonance with the deployment strategy. As an example, in an environment where the human presence is not safe (e.g. biological contamination) it becomes impossible to achieve the results from the grid deployment strategy. Architectures that assume a rigid topology (e.g. one hop communication) become impossible to deploy.

In this paper we presented a new tool to generate realistic network topologies, GenSeN. It allows the user to create new WSN configurations, based on realistic knowledge achieved in previous work performed by the authors. All the results produced by this topology generator are based on real deployment experiences; each placement strategy reflects some of the possible solutions when preparing a WSN scenario.

Based in NS-2 configuration files, GenSeN outputs two *tcl* documents, which can directly be used as part of a network configuration script. GenSeN allows the specification of several parameters, such as different energy levels.

As future work, it would be important to extend the supported deployment strategies. Solutions such as water environment are crucial for certain sensor network applications. Another important extension would be the support of 3D environments. However such extension is more difficult to achieve since the NS-2 core also does not support such feature.

Acknowledgments. The work presented in this paper was partially financed by the IST FP6 CONTENT Network of Excellence (IST-FP6-0384239).

References

1. Estrin, D., et al., “Embedded, Everywhere: A research Agenda for Network Systems of Embedded Computers”, National Research Council Report, 2001.
2. Camilo, T., Rodrigues, A., Sa Silva, J., Boavida, F., “Lessons Learned from a Real Wireless Sensor Network Deployment”, in Proceedings of the Workshop on Performance Control in Wireless Sensor Networks, co-located with Networking 2006 - 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, May 19, 2006;
3. Medina, A., Lakhina, A., Matta, I., Byers, J. “BRITE: Universal Topology Generation from a User's Perspective”. (User Manual) BU-CS-TR-2001-003. April 2005,
4. Calvert, K. Doar, M., Zegura. E. “Modeling Internet Topology” EEE Transactions on Communications, pages 160-163, December 1997.
5. Jin, C., Chen, Q., Jamin, S. “Inet: Internet Topology Generator”. Technical Report Research Report CSE-TR-433-00, University of Michigan at Ann Arbor, 2000.
6. Slijepcevic, S. Potkonjak, M., “Power efficient organization of wireless sensor networks”, In ICC, Helsinki, Finland, June 2001;
7. The USB/LBNL Network Simulator – ns2, http://www.isi.edu/nsnam/ns, 2006;
8. Camilo, T., Rodrigues, A., S. Silva, J., Boavida, F., "Redes de Sensores Sem Fios, considerações sobre a sua instalação em ambiente real" (Wireless Sensor Networks – some Considerations on Deployment in Real Environments), CSMU2006 - Conferência sobre Sistemas Móveis e Ubíquos, Guimarães (Portugal), June-2006 (in Portuguese);
9. Scatterweb, http://www.scatterweb.com/ESB/, 2006;
10. CrossBow, http://www.xbow.com, 2006;
11. I-LENSE Topology Generator (topo_gen) http://www.isi.edu/ilense/software/topo_gen/topo_gen.html, 2005.
12. Arroyo, D., Lee, B., Yu, C., “EMSim: An Extensible Simulation Environment for Studying High Performance Microarchitectures”, in SCI2002: International Conference Challenges Ecoinformatics, 2002.
13. Ali, M., Saif, U., Dunkels, A., Voigt, T., Römer, K., Langendoen, K., Polastre, J. and Uzmi, J., “Medium Access Control Issues in Sensor Networks”, ACM SIGCOMM Computer Communication Review, April 2006.