Emulation of large scale WSN: from real neighbors to imaginary destination

Jovan Radak, Bogdan Pavkovic, Nathalie Mitton, Franck Rousseau, Ivan Stojmenovic

▶ **To cite this version:**

Emulation of large scale WSN: from real neighbors to imaginary destination

Jovan Radak[1] and Bogdan Pavkovic[2] and Nathalie Mitton[1] and Franck Rousseau[2] and Ivan Stojmenovic[3] [†]

[1]*INRIA Lille - Nord Europe, France*
[2]*Grenoble Informatics Laboratory (LIG), University of Grenoble, France*
[3]*SITE, University of Ottawa, Canada*

Setting up large experimental testbeds for wireless sensor networks (WSN) requires access to a huge surface as well as extensive financial and human resources. Due to limited access to such infrastructures, the vast majority of existing theoretical and simulation studies on georouting are not evaluated in real environments. A more affordable approach is needed to provide preliminary insights on network protocols performance. To overcome the need for a large number of sensors required to perform realistic experiments we introduce a novel approach to emulation. We propose to emulate large scale experiment by using a smaller number of core sensor nodes. These nodes are the 1-hop neighborhood of node S, currently holding the packet to forward, and are potential next hops. The destination position is virtual, outside of this real neighborhood. Emulation is performed as follows: *(i)* S transmits the packet over a real wireless channel to the selected forwarding node B, *(ii)* re-map the node B to S and its neighborhood to core nodes, *(iii)* adjust the position of the virtual destination by translating it by $\overrightarrow{BS}$. We repeat these steps until the virtual destination falls into the 1-hop neighborhood of the node currently holding the packet. Compared to real testbeds, our emulation allows testing networks of very high densities and provides unlimited scalability.

Keywords: wireless sensor network, emulation, realistic environment, georouting.

---

1   Introduction

The vast majority of existing WSN testbeds consist of several dozens of nodes. They satisfy the need for experimenting in small scale deployments. Experimental evaluation of network layer protocols intended for large scale WSN raise several difficulties that make them unaffordable to most researchers. First comes to mind the financial cost of a large number of sensor nodes, but it only makes for a small fraction of the problem. The cost of deployment and maintenance, over a large area and in an uncontrolled environment, requiring appropriate human resources, actually represents a huge investment. Thus, using a software-based simulator, like WSNet, OPNet or ns-2, remains the most affordable and popular way to validate WSN protocols. There is a handful of large WSN testbeds, such as SensLAB, the Wisebed federation and GreenOrbs, allowing researchers to test their solutions in a realistic environment and with real hardware.

A compromise between simulation and real testbeds is emulation. Emulation combines real physical experimentation with controlled execution, providing greater control over the tested scenario and environment, as well as a better repeatability. Some assumptions typical to simulations are usually made, or a portion of the system is simulated and interacts with the physical world. In previous work on emulation [4, 5, 1], every physical node is associated with a unique node along a route. The main novelty in our work is the

---

dynamic generation of virtual large scale networks from a small number of real sensor nodes. To achieve this, a single set of nodes is reused at each step of the emulation process, but represents a different subset of the virtual nodes of the large network. Therefore a single physical sensor node may be used several times along a path.

An advantage of our method, even compared to full size experimentation, is the ability to work with very dense networks. We can choose an arbitrary maximum density by placing enough nodes in the 1-hop neighborhood. At each step, only a subset of these real nodes might be effectively used. The ultimate advantage of this approach is its unlimited scalability: it can provide virtually unbounded network scenarios. We will obviously sacrifice accuracy and realism up to certain extent, but are likely to gain much more insight than what is obtained from simulation. Our method can be applied on existing testbeds (from 50 to 1000 nodes) to emulate the performance of a million plus WSN.

2 Emulation

In this section we describe the general idea of our original emulation proposition [2] and show some of its weaknesses. Then we will describe our ongoing effort on emulation and its improvements over the initial proposition.

2.1 Using a real one-hop neighborhood

Every available physical nodes is provided with its geographic location as well as the location all of its neighbors. One of the nodes, placed at the center of a circle of radius r (approximative range), plays the role of the source, node S in Fig. 1. The rest of the available sensors are placed randomly in the circle, creating the 1-hop neighborhood of S. The destination node D is a virtual node, placed outside of this 1-hop neighborhood. Using the real physical layer and the emulated routing algorithm, S selects B as its next hop. After successful transmission of a packet from node S to node B, the virtual destination is translated by vector $\overrightarrow{BS}$ to a new virtual destination D'. This process continues until the current position of the virtual destination falls inside the radius of the 1-hop neighborhood, or when the routing algorithm fails. We can imagine this emulation process as if the 1-hop real neighborhood would translate together with the forwarded message from source to destination through a large network.
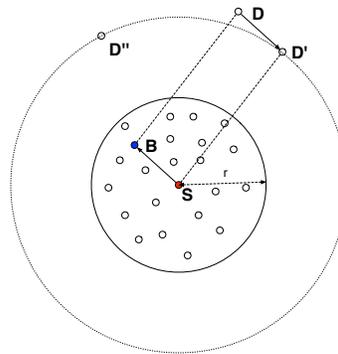

Fig. 1: One forwarding step during emulation

One problem with this approach is that the neighborhood around S does not change along the emulated path. To overcome this, D' is further rotated by a random angle to position D'' in Fig. 1). This will change the actual candidate neighborhood configuration at each step, which is closer to a real experimental setup. But we cannot expect full randomness of the neighborhood with such a rotation.

2.2 One-hop hexagonal grid neighborhood

The source node S is still positioned at the center of a circle of radius r. Its 1-hop neighborhood consists of 43 WSN430 sensor nodes that are placed following an hexagonal pattern in this circle, as shown in Fig. 2(a).

(a) Placement and enumeration of the sensors

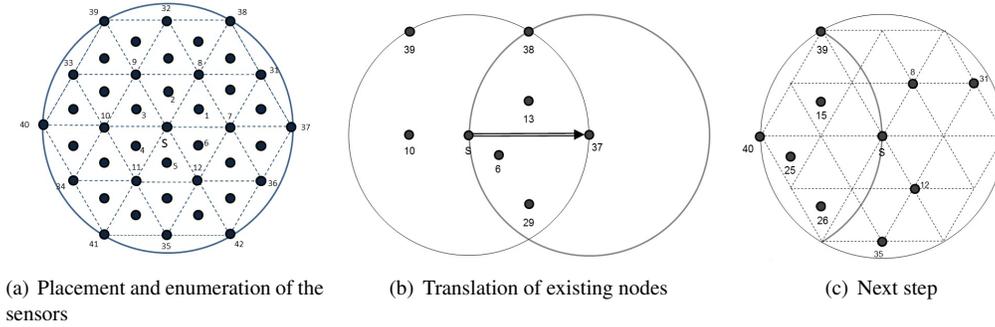(b) Translation of existing nodes

(c) Next step

Fig. 2: Hexagonal grid used for the placement of sensors.

The placement of nodes on a hexagonal grid ensures that in two successive steps, each node which lies at the intersection of successive neighborhoods translates to one of the nodes of the same grid — due to central and axial symmetry of the hexagonal grid structure. Although this structure is highly regular, it still offers significant variety in node placement and distance to central node *S*.

To overcome the limitation of the previous proposition, and generate a wide range of different neighborhoods, we do not use all the nodes of the 1-hop environment at every step. Instead, we are using the nodes that lie at the intersection of two successive steps — nodes that are at the intersection of circles centered at the source node *S* and at the chosen forwarding node — and an arbitrary subset of the rest of the nodes. In this way we provide more realistic emulation, control the density of the large scale network that is being emulated and emulate obstacles.

An example of this emulation process is shown in Fig. 2. Let us assume that the message in transit is currently held by source *S* and that the routing algorithm selects node 37 in Fig. 2(a) as the next hop. *B* is supposed to be the placement of the source node in the next step of the emulation process, thus drawing the circle of radius *r* centered at *B* will give us the common area in two successive steps as shown in Fig. 2(b). In the next step, *B* translates to *S*, so in order to keep the relative positioning of all nodes in the common area we are translating them by $\overrightarrow{BS}$, virtually moving the circle of radius *r* from *B* to *S*. Due to the properties of symmetry of the hexagonal grid structure, each node will be exactly mapped to one of the existing nodes, in Fig. 2(c): $13 \mapsto 15$, $6 \mapsto 25$, $29 \mapsto 26$, $38 \mapsto 39$, $S \mapsto 40$ and $37 \mapsto S$. In addition to common nodes with the previous hop, any subset of nodes among the remaining ones can be chosen for the next step, using any appropriate algorithm. These nodes are shown outside of intersection of two successive steps on Fig. 2(c). In comparison with the previous solution [2], we do not apply rotation of the virtual destination because at each step we pseudo-randomly choose a subset of the nodes which are going to be used in the next iteration, ensuring that different edges of the graph will be used.

3 Emulation and simulation

As a proof of concept of this approach to emulation and in order to emphasize its benefits, we compare it to the simulation results with the same scenarios. Both simulation and emulation are run on the graph generated using the MIN-DPA algorithm [3]. Results are obtained using the WSNet simulator, running the simulation of emulation and pure simulation of the large scale network generated using MIN-DPA. For both simulation and emulation all nodes first perform an initialization phase where they broadcast 1 024 messages, grouped in 8 rounds, in order to evaluate link quality i.e. to measure the ETX link metric. Link quality statistics are gathered in the source node *S* for the case of emulation. At this point we have the probability of reception for each pair of nodes, and we can use these data in later experiments.

We proceed by evaluating the performances of different geographical routing algorithms based on ETX measurements. Similarly to our previous work [2] we run the following routing algorithms: XTC [6], GARE, COP_GARE and additionally LEARN-G. We compare the results obtained by emulation and simulation. For each run we have 4 source nodes, located in the 4 corners of the square network. Routing is performed across the diagonals to reach the destination situated in the opposite corner. A parameter of

interest in the total number of hops needed to reach the destination.

MIN-DPA algorithm generates connected graphs with no crescent holes. This results in 100% success rate for all routing schemes. All of them are greedy-like with no face recovery phase.

The comparison of results for emulation and simulation are shown in Fig. 3. They are similar to those presented in previous work [2] both for the simulation on the graph and the simulation of the emulation —process in which we simulate the behavior of emulation. The XTC algorithm, using the shortest links, shows the largest number of hops per routing, while LEARN-G, favoring longer links, gives the smallest number of hops.

The performances of GARE and COP_GARE are between LEARN-G and GARE algorithms, since they are both using mid-sized edges. The biggest difference between simulation and emulation is for the case of LEARN-G: it has a slightly smaller number of hops per route for the case of simulation which can be explained by the fact that the algorithm was not constrained to 1-hop neighborhood, as it was the case with emulation, and could have also used neighbors outside the 1-hop neighborhood.
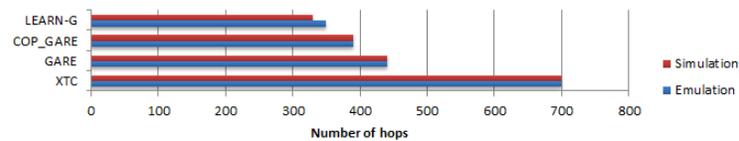

Fig. 3: Average number of hops for various routing algorithms

4   Conclusion

We have presented a new approach to emulation, paving the way for experimentation on realistic unbounded wireless sensor networks at reasonable cost, with any node density and obstacles. In our experiments we run routing algorithms that use ETX as one of the parameters of their metric. A set of greedy-like algorithms was evaluated using simulation of emulation and plain simulation using WSNet in order to compare the results. We have shown the consistency of the results obtained running the simulations with hexagonal structure and with regular simulation using the same graph generated using MIN-DPA: the structure we proposed can give accurate results. Emulated networks can be of any size, with a wide range of node number, distribution and density. Human effort and code complexity remain reasonable, still allowing large scale experimentation.

References

[1] Glenn Judd and Peter Steenkiste. Using emulation to understand and improve wireless networks and applications. In *NSDI*, 2005.

[2] Milan Lukic, Bogdan Pavkovic, Nathalie Mitton, and Ivan Stojmenovic. Greedy geographic routing algorithms in real environment. In *MSN*, 2009.

[3] Furuzan Atay Onat, Ivan Stojmenovic, and Halim Yanikomeroglu. Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks. *Pervasive and Mobile Computing*, 4, 2008.

[4] Ahmed Sobeih, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Honghai Zhang, Wei-Peng Chen, Hung-Ying Tyan, and Hyuk Lim. J-sim: a simulation and emulation environment for wireless sensor networks. *Wireless Communications, IEEE*, 13, 2006.

[5] Joseph Sventek, Alasdair Maclean, Ross McIlroy, and Grzegorz Milos. Xenotiny: Emulating wireless sensor networks on xen. Technical report, University of Glasgow, Department of Computing Science, 2008.

[6] Roger Wattenhofer and Aaron Zollinger. Xtc: A practical topology control algorithm for ad-hoc networks. In *IPDPS*, 2004.