Demo Abstract: Distributed Topology Control in Wireless Sensor Network

Luca Casartelli
Libero Istituto Universitario Carlo Cattaneo

Matteo Cesana
Politecnico di Milano

Demo Abstract: Distributed Topology Control in Wireless Sensor Network

S. Brambilla, L. Casartelli, M. Cesana*
Dipartimento Elettronica ed Informazione
Politecnico di Milano
Piazza L. da Vinci 32
20133, Milano, Italy
Email*: cesana@elet.polimi.it

***Abstract*— The choice of transmission power levels adopted in Wireless Sensor Networks (WSNs) is critical to determine the performance of the network itself in terms of energy efficiency, connectivity and spatial reuse, since it has direct impact on the physical network topology. In this work, we demonstrate a lightweight and fully distributed solution to adaptively tune the transmission power of MicaZ in order to match local connectivity constraints.**

I. Background

Network topology has huge impact on the efficiency of Wireless Sensor Networks: at the MAC layer, the more connected is the network the higher is the collision probability, whereas the routing layer requires high connectivity degrees to set up effective routes. Hence, the design of effective distributed topology control protocols for WSN is a crucial issue which might determine the success of sensor network technology itself [1]. Given the peculiarity of the WSN domain, topology control protocols must be fully or partially distributed and highly flexible and adaptive to cope with high network variability due to node mobility, wireless link quality fluctuation, or activity cycling.

In this demo, we are interested in topology control as a way to determining the sensors'degree K, i.e. the number of neighbors directly connected to a given sensor through a bidirectional wireless link [2]. To this end, the Topology Control (TC) algorithm we want to demonstrate aims to provide a minimum value K of bidirectional links for each sensor node. Moreover, a cooperation scheme is introduced to tackle critical connectivity issues that may appear in real network scenarios. More in details, the TC algorithm is composed of two distinct phases: the Neighbor Discovery (ND) and the Topology Update (TU). Both phases are performed periodically on a constant time basis, without making any assumption on how TU and ND are scheduled on different nodes.

In the ND phase, sensors broadcasts beacon messages carrying local ID, the list of current neighbors N and the current transmission power level. When a node overhears a beacon it stores the information about the sender and creates a list of overheard neighbors ordered on the basis of the transmission power level contained in the corresponding beacon.

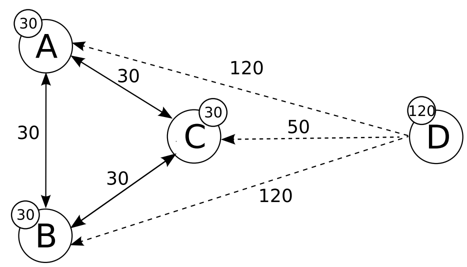

Fig. 1. Operation of the topology control algorithm: a simple example ($K = 2$).

During the TU each sensors has to decide which power level to use for transmission. Such decision is based on the information stored in the local neighborhood list. Each node computes how many neighbors it has collected during the discovery phase. If they are less than K, it increases the transmission range by a factor ρ_{inc} defined as protocol parameter. Otherwise, if the number of neighbors is equal or greater than K, the transmission power is regulated to reach at most the K^{th} neighbor. This way, a node is free to adaptively tune its range to cover exactly K neighbors.

The protocol described so far (referred to as *basic protocol* in throughout this paper) is indeed successful in maintaining the desired neighbor degree on each node and it enables saving a large amount of energy if compared with protocols without topology control [3]. Nevertheless, it still shows some drawbacks that may negatively impact overall performance. This undesired behavior is depicted in Figure I. There are three nodes, namely A, B and C, which have already reached the desired local connectivity ($K = 2$ in this case). However, there exists a further node D which instead needs to find K neighbors. Unfortunately, regardless how long its transmission range is, it is unable to connect to any node, since all other nodes have already K neighbors and are therefore unwilling to extend their range to include D. According to the *basic protocol*, D would end up transmitting at maximum power, thus consuming high power and creating significant interference to other communications.

To avoid this behavior, an enhanced version of the topology protocol is introduced which leverages the list neighbor provided by each node in its beacons. During the discovery phase, when a node r receives a beacon from s, it computes the size of s's neighborhood ($\mathcal{N}_s$) and if it is lower than K, s is marked as *critical*. In the update phase, critical nodes are included as neighbors and transmission range is modified accordingly. We name this version of the topology control protocol *list-based protocol*.

II. Demo Description

To demonstrate the behavior of topology control algorithms, we use 16 MicaZ nodes [5] deployed in the demo area and featuring ChipCon CC2420 radio transceivers [6]. Each of the sensors is geared with the topology control functionalities described in the previous section and properly implemented in TinyOS-1.x. As for the MAC layer, the sensors run TinyOS B-MAC without low power listening implementation.

Moreover, since it is very unpractical to manually download data sensor by sensor at the end of each experiment, we implement an automatic procedure to collect at a sink node all the data stored during the experiment by all the other sensors. To this end, one MicaZ node acts as information sink and is directly connected through a MIB510 Serial Gateway to a PC running Linux distribution Debian with 2.6.18 kernel version. Each sensor collects and stores periodical samples of information during the experiment including the list of perceived neighbors (and the corresponding power levels) and the current transmissions power. Upon completing the experiment, each sensor searches for a path to the sink using the MintRoute routing protocol [4] and then sends to the sink all these information samples.

The sink sensor passes such information to the PC which runs a Java filter, returning the overall performance measures used to evaluate the topology control solutions. Moreover, the Java tool implements also a query mechanism based on a diffusion protocol to force the sink node to request missing information that may get lost during the collection phase.

We measure the following performance metrics:

- *local connectivity*: local connectivity of any sensor in terms of logical neighbors, i.e. those neighbors connected through symmetric link, and physical neighbors, i.e. all nodes reached by sensor beacons;
- *network connectivity*: ratio between the number of vertexes of the largest bidirectional links based connected sub-graph and the total number of nodes;
- *average transmission power* and transmission power distribution.

The Java host application provides also a GUI that graphically shows the behavior of the measured statistics over the experiment time. Moreover, the GUI is also able to display the communication graph describing actual vertexes and edges of the network.

Two set of demonstrative experiment will be carried out:

- *Cold Start Experiment:* nodes are switched on from scratch and run the specific topology control algorithm for a given period of time (100s). After that, they move onto the second phase devoted to data collection and data elaboration, during which experiment data is collected at the sink and elaborated by the Java application.
- *Stress-Test Experiments:* starting from a stable topology, nodes will be switched off and/or moved to other locations to test the capability of the topology control algorithm to react to topology changes.

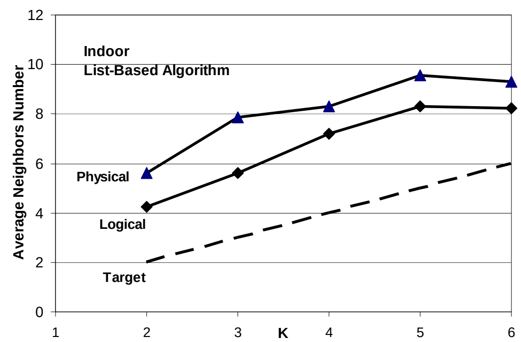

Fig. 2. Physical and logical neighbors against K.

III. A Flavor of results

To give an idea of the type of results we gather, we have run an indoor experiment at the Advanced Network Technology Lab (ANT-LAb) of the Politecnico di Milano. Figure 2 reports the measured number of logical and physical neighbors at the end of the experiment against the target value K for the *list-based* topology control approach. Notably, the list-based algorithm provides a number of logical neighbors slightly higher than the target parameter K. Such difference in excess is due to the "price of cooperation", that is, the fact that the cooperative approach forces a subset of nodes to increase their transmission power to help critical neighbors. Other results will be gathered and commented during the demonstration.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, *A Survey on Sensor Networks*, IEEE Comm. Mag., Aug. 2002, Vol. 40, Issue: 8, pp. 102–114.

[2] F. Xue, P. R. Kumar, *The Number of Neighbors Needed for Connectivity of Wireless Networks*, Wireless Networks, March 2004, 10(2).

[3] M. Kubisch, H. Karl, A. Wolisz, L.C. Zhong, J. Rabaey, *Distributed algorithms for transmission power control in wireless sensor networks*, in Proc. of IEEE WCNC 2003.

[4] A. Woo, T. Tong, D. Culler, *Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks*, in Proc. of ACM Sensys 2003.

[5] www.xbow.com

[6] www.chipcon.com