

# Simulating Synchronized Calling with a Wireless Sensor Network

Kenneth D. Frampton, Efosa Ojomo, Praveen Mudindi, and Isaac Amundson  
Department of Mechanical Engineering  
Vanderbilt University  
Nashville, TN, USA

## ***Abstract.***

Inspired in part by last summer's emergence of cicada brood X, we set out to program a wireless sensor network to mimic the synchronized vocalizations exhibited by many insect and animal species. There are a number of different theories, supported by numerous studies, that attempt to explain such naturally occurring synchronized behaviors. These behaviors may have evolved cooperatively in order to maximize signal loudness, to confuse predators, or to improve call features that attract potential mates. Alternatively, synchronized calling may have evolved competitively in order to mask or jam the calls of nearby animals. Whichever theory is true, it is clear that these behavior patterns are complex and offer an interesting inspiration for group behaviors.

Working from this inspiration, we set out to program the nodes in a distributed network to mimic this natural behavior. Networks consisting of nodes that communicate with each other but act independently according to simple rules are becoming increasingly popular, but there is a great deal that we do not yet know about the group behavior of such systems. In addition to being a lot of fun, the synchronized calling experiment adds to our understanding of the behavior of this kind of network.

We adapted a group of about 15-20 Mica II Motes so that they would behave in the manor described previously. Mica II Motes, designed by computer scientists at UC Berkeley and manufactured by Crossbow Inc., are small microprocessors equipped with wireless communications. These devices can receive expansion boards that allow them to perform various tasks such as environmental monitoring, security, and tracking. In our case, we included an expansion board with a microphone and a buzzer.

In order to mimic synchronized calling behaviors we began by programming a single leader called an Alpha node. This device was programmed to begin calling (i.e. buzzing) with an arbitrary call length and arbitrary frequency. The Alpha node also called at this rate regardless of any other calling in its vicinity. The remainder of the devices, referred to as Beta nodes, were programmed to listen with their microphones. When a Beta node receives a call that is sufficiently loud, it estimates the call's duration and frequency. Once this is done the node begins calling in synch with the detected call.

Although this behavioral algorithm is quite simple, it produces some interesting group behaviors. When all is quiet and an Alpha node begins calling, only those Beta nodes close to the Alpha node hear the call and respond. Then, as more Betas swell the chorus, nodes further away hear the call and join in. In this fashion, the synchronized calling gradually spreads concentrically out from the Alpha node until all nodes were synchronized.

A second interesting behavior occurred when a Beta node would "hiccup" and call out of synch with its neighbors. Such hiccups are attributable to measurement noise, operating system jitter and other factors. Occasionally, when such a hiccup occurred, neighboring nodes would resynchronize to the errant node. These transients would normally quickly disappear as the wayward group resynchronizes with the larger group.

The most interesting behavior pattern appeared when we introduced an Omega node: a node that was programmed identically to the Alpha node but set to a different duration and frequency. When introduced into the array, the Omega node would begin to attract neighboring nodes to its call cycle. Unlike the hiccup case, however, this group does not resynchronize with the original group. Rather, the Omega node eventually recruits a large number of nodes to its calling cycle until a "balance of power" is reached. The eventual balance between the two groups depends strongly on the arrangement of the sensors.

While this is a rather whimsical application of a sensor network, it demonstrates the unique system behaviors that can arise in truly distributed processing. Even when nodes follow very simple rules, the behavior of the group can be quite complex. Although this work has not likely improved our knowledge on synchronized calling in nature, it demonstrates the types of complex behavior patterns that will be important for future developments in sensor networks.