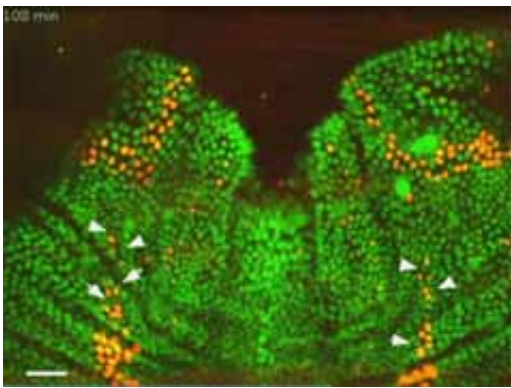# Fruit fly nervous system provides new solution to fundamental computer network problem

January 13 2011

In this confocoal miscroscope image of the pupal stage of fruit fly development, nerve cells that self-select to become sensory organ precursors (SOPs) are identified by arrows. These cells send chemical signals to neighboring cells, blocking them from becoming SOPs and causing them to fluoresce red in the image.

(PhysOrg.com) -- The fruit fly has evolved a method for arranging the tiny, hair-like structures it uses to feel and hear the world that's so efficient a team of scientists in Israel and at Carnegie Mellon University says it could be used to more effectively deploy wireless sensor networks and other distributed computing applications.

With a minimum of communication and without advance knowledge of

how they are connected with each other, the cells in the fly's developing [nervous system](#) manage to organize themselves so that a small number of cells serve as leaders that provide direct connections with every other nerve cell, said author Ziv Bar-Joseph, associate professor of machine learning at Carnegie Mellon University.

The result, the researchers report in the Jan. 14 edition of the journal *Science*, is the same sort of scheme used to manage the distributed [computer networks](#) that perform such everyday tasks as searching the Web or controlling an airplane in flight. But the method used by the fly's nervous system to organize itself is much simpler and more robust than anything humans have concocted.

"It is such a simple and intuitive solution, I can't believe we did not think of this 25 years ago," said co-author Noga Alon, a [mathematician](#) and computer scientist at Tel Aviv University and the Institute for Advanced Study in Princeton, N.J.

Bar-Joseph, Alon and their co-authors — Yehuda Afek of Tel Aviv University and Naama Barkai, Eran Hornstein and Omer Barad of the Weizmann Institute of Science in Rehovot, Israel — used the insights gained from [fruit flies](#) to design a new distributed computing algorithm. They found it has qualities that make it particularly well suited for networks in which the number and position of the nodes is not completely certain. These include [wireless sensor networks](#), such as environmental monitoring, where sensors are dispersed in a lake or waterway, or systems for controlling swarms of robots.

"Computational and mathematical models have long been used by scientists to analyze biological systems," said Bar-Joseph, a member of the Lane Center for Computational Biology in Carnegie Mellon's School of Computer Science. "Here we've reversed the strategy, studying a biological system to solve a long-standing computer science problem."

Today's large-scale computer systems and the nervous system of a fly both take a distributive approach to performing tasks. Though the thousands or even millions of processors in a computing system and the millions of cells in a fly's nervous system must work together to complete a task, none of the elements need to have complete knowledge of what's going on, and the systems must function despite failures by individual elements.

In the computing world, one step toward creating this distributive system is to find a small set of processors that can be used to rapidly communicate with the rest of the processors in the network — what graph theorists call a maximal independent set (MIS). Every processor in such a network is either a leader (a member of the MIS) or is connected to a leader, but the leaders are not interconnected.

A similar arrangement occurs in the fruit fly, which uses tiny bristles to sense the outside world. Each bristle develops from a nerve cell, called a sensory organ precursor (SOP), which connects to adjoining [nerve cells](#), but does not connect with other SOPs.

For three decades, computer scientists have puzzled over how processors in a network can best elect an MIS. The common solutions use a probabilistic method — similar to rolling dice — in which some processors identify themselves as leaders, based in part on how many connections they have with other processors. Processors connected to these self-selected leaders take themselves out of the running and, in subsequent rounds, additional processors self-select themselves and the processors connected to them take themselves out of the running. At each round, the chances of any processor joining the MIS (becoming a leader) increases as a function of the number of its connections.

This selection process is rapid, Bar-Joseph said, but it entails lots of complicated messages being sent back and forth across the network, and

it requires that all of the processors know in advance how they are connected in the network. That can be a problem for applications such as wireless sensor networks, where sensors might be distributed randomly and all might not be within communication range of each other.

During the larval and pupal stages of a fly's development, the nervous system also uses a probabilistic method to select the cells that will become SOPs. In the fly, however, the cells have no information about how they are connected to each other. As various cells self-select themselves as SOPs, they send out chemical signals to neighboring cells that inhibit those cells from also becoming SOPs. This process continues for three hours, until all of the cells are either SOPs or are neighbors to an SOP, and the fly emerges from the pupal stage.

In the fly, Bar-Joseph noted, the probability that any cell will self-select increases not as a function of connections, as in the typical MIS algorithm for computer networks, but as a function of time. The method does not require advance knowledge of how the cells are arranged. The communication between cells is as simple as can be.

The researchers created a computer algorithm based on the fly's approach and proved that it provides a fast solution to the MIS problem. "The run time was slightly greater than current approaches, but the biological approach is efficient and more robust because it doesn't require so many assumptions," Bar-Joseph said. "This makes the solution applicable to many more applications."

Provided by Carnegie Mellon University