SELF-ORGANIZING ROUTING ALGORITHM FOR WIRELESS SENSORS NETWORKS (WSN) USING ANT COLONY OPTIMIZATION (ACO) WITH TINYOS

Nuria Gómez Blas, Luis F. de Mingo, Levon Aslanyan, Vladimir Ryazanov

Abstract*: This paper describes the basic tools to work with wireless sensors. TinyOShas a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools – all of which can be used as-is or be further refined for a custom application. TinyOS was originally developed as a research project at the University of California Berkeley, but has since grown to have an international community of developers and users.Some algorithms concerning packet routing are shown. In-car entertainment systems can be based on wireless sensors in order to obtain information from Internet, but routing protocols must be implemented in order to avoid bottleneck problems. Ant Colony algorithms are really useful in such cases, therefore they can be embedded into the sensors to perform such routing task.*

Keywords*: Mobile ad-hoc networks, ant colony optimization, routing protocols, simulation, TinyOS.*

ACM Classification Keywords*:A.0 General Literature - Conference proceedings, I.6. Simulation and Modelling, I.2.8 Problem Solving, I.2.11 Distributed Artificial Intelligence, H.5.2 Information interfaces and presentation: Miscellaneous.*

Preliminaries

Ant Colony Optimization (ACO) [19, 20] technique is anoptimization technique to solve optimization problem. It has been developed for combinatorial optimization problems. ACO are multi-agent system in which the behaviour of each single agent, called ant, is inspired by the behaviour of real ants. ACO has been successfully employed to combinatorial optimization problems such as maximum loadability in voltage control study, loss minimization in distribution networks, unit commitment problem, multiobjective reactive power compensation, and complex multi-stage decision problem.

Figure 1 shows the behavior of an ant colony. There is a path, along which ants are walking, for example from food source A to the nest E, and vice versa. Suddenly an obstacle appears and the path is cut off. So at position B the ants walking from A to E (or at position D those walking in the opposite direction) have to decide whether to turn right or left. The path of ants followed is shown in figure 1.Because path BCD is shorter than BHD, the first ant following it will reach D before the first ant following path BHD which is shown in figure 1.c. The result is that an ant returning from E to D will find a stronger trail on path DCB, caused by the half of all the ants that by chance decided to approach the obstacle via DCBA and by the already arrived ones coming via BCD: they will therefore prefer (in probability) path DCB to path DHB. As a consequence, the number of ants following path BCD per unit of time will be higher than the number of ants following EHD. This causes the quantity of pheromone on the shorter path to grow faster than on the longer one, and therefore the probability with which any single ant chooses the path to follow is quickly biased toward the shorter one. The final result is that very quickly all ants will choose the shorter path.

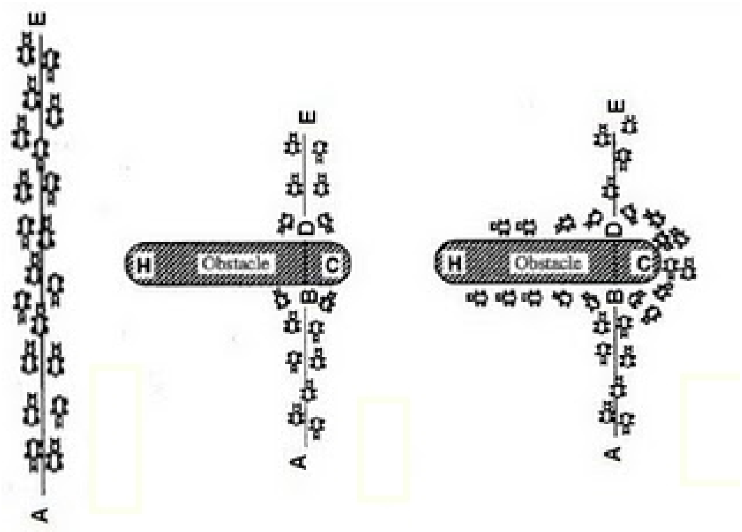


Figure 1.- Ant Colony Behaviour.

Ants use the environment as a medium of communication. They exchange information indirectly by depositing pheromones, all detailing the status of their "work". The information exchanged has a local scope, only an ant located where the pheromones were left has a notion of them. This system is called "Stigmergy" and occurs in many social animal societies (it has been studied in the case of the construction of pillars in the nests of termites). The mechanism to solve a problem too complex to be addressed by single ants is a good example of a self-organized system.

The algorithm is derived from the study of real ant colonies. Therefore the system is called as Ant System (AS) [20] and the algorithms as Ant algorithms. The use of artificial ant colonies as an optimization tool, will have some major differences with a real (natural ant) one:

1. artificial ants will have some memory,

2. they will not be completely blind,

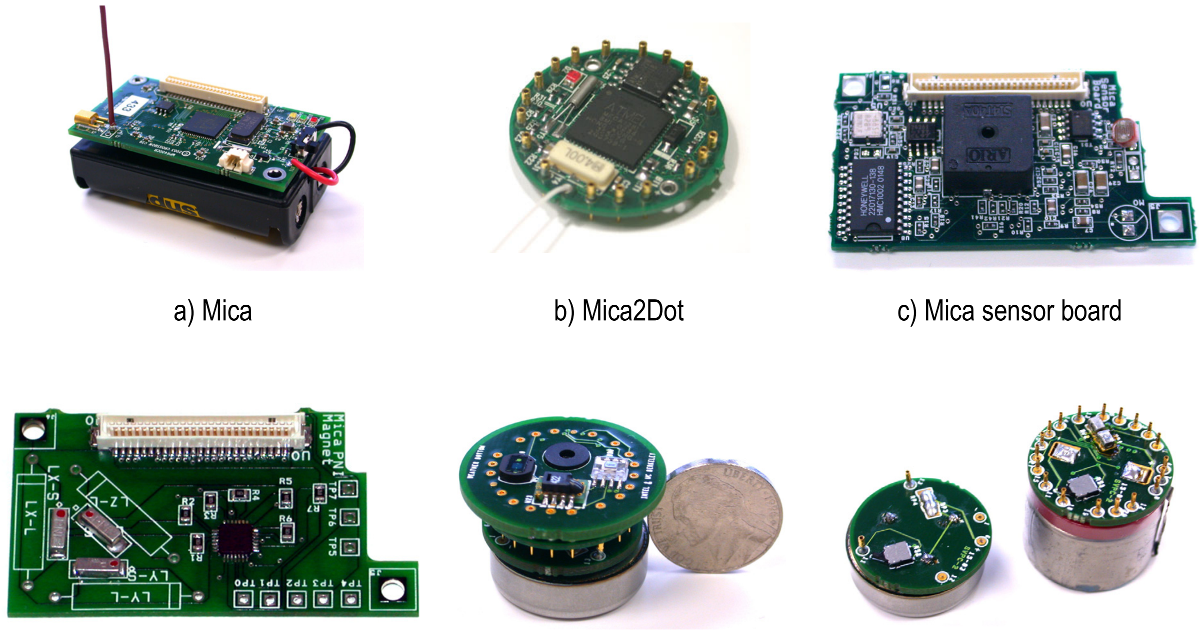
3. they will live in an environment where time is discrete.

Ant colony optimization is a major breakthrough in Engineering as well as non engineering applications.

Wireless Sensors Networks Tools

Wireless Sensors Networks (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, and to cooperatively pass their data through the network. The more modern networks are bi-directional, enabling also to control the activity of the sensors. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer application, such as industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

The WSN is built of "nodes" - from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts, see figure 2: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding [9, 10, 11].



a) Mica b) Mica2Dot c) Mica sensor board

d) PNI Magnetometer/Compass e) Dot sensor board f) Power supplies

Figure 2.-Hardware implementation of different motes and available sensors and power supplies to connect to.

TinyOS

TinyOS[1, 16] is an event-driven operating system designed for sensor network nodes that have very limited resources. TinyOS, is used, for example, on the MICA motes (see figure 2), which are small wireless sensor nodes. TinyOS has extensive networking support, and this support includes technically excellent protocol designs which have become de facto standards, or in some cases, parts of Internet standards. This support has been in part due to TinyOS's use as a platform by many leading low-power wireless research groups, who have then released their code for general use and supported it well. The TinyOS net2 Working Group is responsible for adding, improving, and maintaining TinyOS's network protocols.

TinyOS supports low duty cycle operation through low-power link layers. Rather than keep the radio always on, TinyOS turns the radio on periodically (e.g., every few hundred ms) to check if there is a packet to receive. This enables the network to appear "always on" yet support sub-1% duty cycles: the basic tradeoff is that communication has higher latency.TinyOS supports multihop, network-wide sub-millisecond time synchronization through the Flooding Time Synchronization Protocol, developed by researchers at Vanderbilt University.

Data collection protocols build a self-organizing, self-repairing routing topology to data collection points known as "roots." Typically these roots are connected to a PC or other device, such that the data collected can be stored in a database for later use. Collection protocols send data in only one direction (towards a root): they do not support messages to arbitrary nodes in the network. TinyOS's standard collection protocol, the Collection Tree Protocol (CTP), is highly efficient and robust: it continues to deliver data even after large numbers of node failures and has emerged as the gold standard against which other routing protocols are measured.

Data dissemination protocols reliably deliver a piece of data to every node in a network. TinyOS supports three dissemination protocols: Drip, DIP, and DHV [13, 14, 15]. These three protocols represent a gradual evolution towards more efficient algorithms. Generally speaking, applications should use DHV.

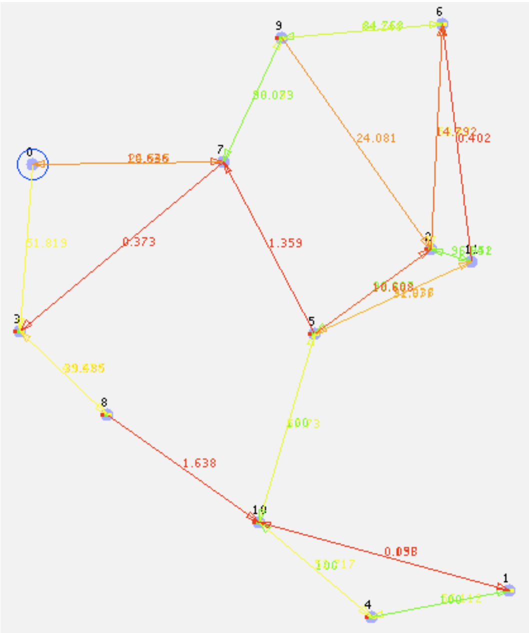
TinyOS includes support for reprogramming a multihop wireless network over the air with the Deluge protocol. A Deluge-enabled network supports having multiple binaries in the network at once: a command line tool can instruct the network to change programs. This operation takes a short while as the nodes reprogram themselves.

All of the above protocols are subjects of a long literature of research and publications, such that there is extensive information in how they work. They are all designed to work on top of low power link layers [12].

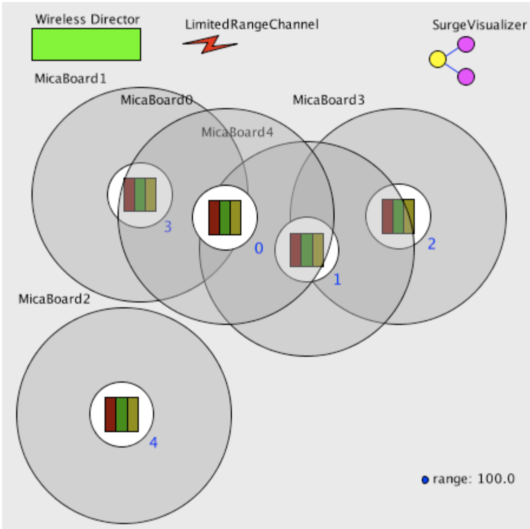
Tinyviz, Viptos and Ptolemy II

Viptos (Visual Ptolemy and TinyOS) [2, 4] is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. Viptos allows developers to create block and arrow diagrams, see figure 3, to construct TinyOS [1] programs from any standard library of nesC/TinyOS components. The tool automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target hardware. Viptos is based on TOSSIM [1] and Ptolemy II [3]. TOSSIM is an interrupt-level simulator for TinyOS programs. It runs actual TinyOS code but provides software replacements for the simulated hardware and models network interaction at the bit or packet level. Ptolemy II [7, 8] is a graphical software system for modelling, simulation, and design of concurrent, real-time, embedded systems. Ptolemy II focuses on assembly of concurrent components with well-defined models of computation that govern the interaction between components. While TOSSIM only allows simulation of homogeneous networks where each node runs the same program, Viptos supports simulation of heterogeneous networks where each node may run a different program. Viptos simulations may also include non-TinyOS-based wireless nodes. The developer can easily switch to different channel models and change other parts of the simulated environment, such as creating models to generate simulated traffic on the wireless network.

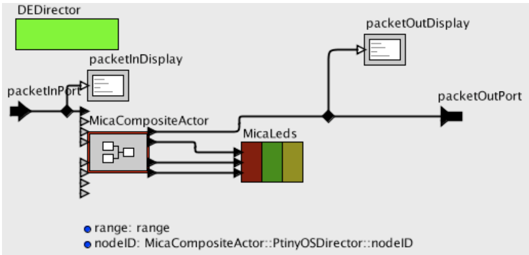
Viptos inherits the actor-oriented modelling environment of Ptolemy II [5, 6], which allows the developer to use different models of computation at each level of simulation. At the lowest level, Viptos uses the discrete-event scheduler of TOSSIM to model the interaction between the CPU and TinyOS code that runs on it. At the next highest level, Viptos uses the discrete-event scheduler of Ptolemy II to model interaction with mote hardware, such as the radio and sensors. This level is then embedded within VisualSense to allow modelling of the wireless channels to simulate packet loss, corruption, delay, etc. The user can also model and simulate other aspects of the physical environment including those detected by the sensors (e.g., light, temperature, etc.), terrain, etc.



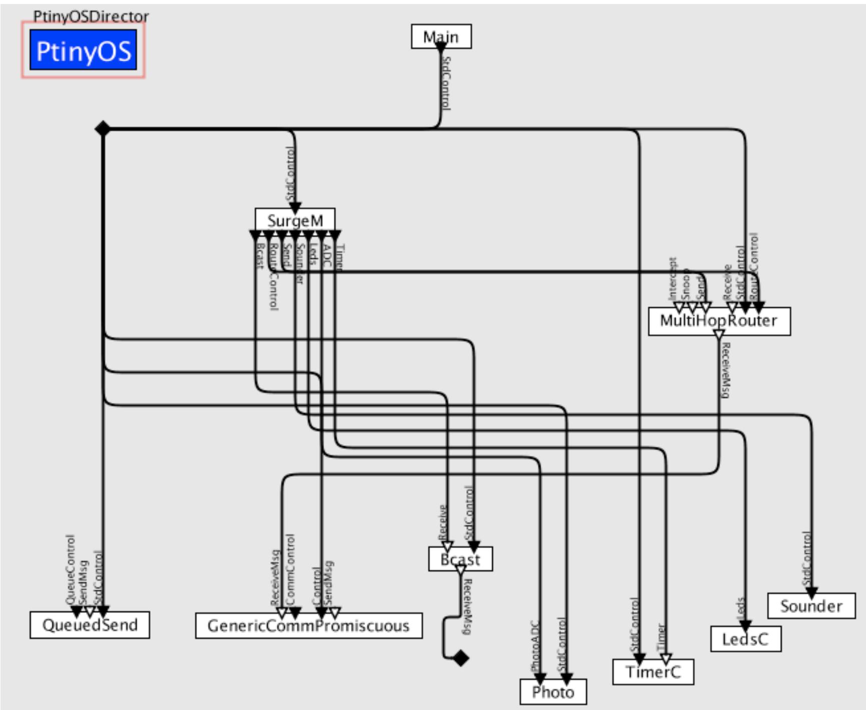
a) TiniViz simulation of motes showin radio link power.



b) Distribution of motes in space.



c) Mote behaviour.



d) Mote components and interfaces.

Figure 3.- TiniViz simulation of motes with radio links a), and Viptos simulacion at different abstraction levels b), c) and d).

TinyViz [1] is a Java-based GUI that allows you to visualize and control the simulation as it runs, see figure 3.a), inspecting debug messages, radio and UART packets, and so forth. The simulation provides several mechanisms for interacting with the network; packet traffic can be monitored, packets can be statically or dynamically injected into the network.

Ant colony self-organizing routing algorithms

Ant colony algorithms were first proposed by Dorigo et al as a multi-agent approach to difficult combinatorial optimization problems like the traveling salesman problem (TSP), figure 4, and the quadratic assignment problem (QAP), and later introduced the ACO meta-heuristic.

Figure 4.- Solved TSP problem using ant colony optimization.

There are two types of ants applied in the algorithm, forward ants and backward ants. Forward ants, whose main actions are exploring the path and collecting the information from the source nodes to destination node, have the same number as the source nodes [21]. The paths that forward ants travel will construct a tree when they merge into each other or reach the destination and data is transmitted along the tree paths, see figure 5.

POSANT Routing Algorithm [19, 20] is ant colony optimization based routing algorithm which uses location information to improve its efficiency. POSANT is able to find optimum or nearly optimum routes when a given network contains nodes.*Zone based Routing Algorithm using Cluster*. Concept of clustering needs grouping of nodes in the network. This grouping depends upon transmission range and number of hop in a group. Each node group will have a group head called Cluster head having the responsibility of communication among its member nodes and other cluster heads. Cluster head should contain address of its member nodes as well as that of other cluster heads. Member nodes need to store address information of their cluster head and neighbor nodes. When information needs to pass from one node to another, member node sends this information to its corresponding cluster head, which decides whether the destination is a member or not.

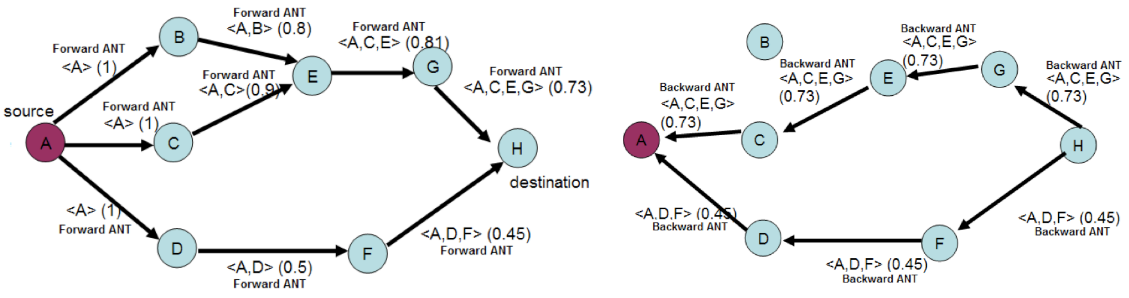


Figure 5.- Ant colony routing in mobile ad-hoc networks (MANET) protocol [17].

Ant Colony Routing Algorithm with Zones [18]. Concept of Ant Colony algorithm is merged with zone based (clustering) algorithm to form ant colony routing algorithm with zones. This algorithm will provide advantage of both ant colony and zone based algorithm. Like ant colony algorithm, here we need not store large routing tables in nodes, we need to store only neighboring node information and previous traversed node information. As nodes in mobile ad-hoc network will have memory of small storage capacity, it would be tough to store large routing table inside each node.

A local routing, instead of storing the whole network graph, will be more suitable in order to keep track of the information going to a destination node.

The local routing table in every node/mote keeps the following information:

A list of neighbourhood nodes/motes that have internet connection, see figure 6. This table is build using a discovery ant that every node will run, when the ant reaches the internet sink a backward ant will be sent back to the source node that updates the probability and lookup table of nodes. These discovery ants start if there is no internet connection at regular time intervals.

In order to be able to sent back data packets, the MAC address or ID of the source node must be keep in the path of the route to the internet sink. That is, every node/mote stores the pair:

the MAC/ID of the source of every transmitted packet (to be able to sent data back to the source),

and the MAC/ID of the connected node/mote of transmitted packet (to sent data back).

Data packets are sent using the Internet lookup table, according to the propabiliy of the node. When the echo information pass a node and reaches the source, then the probabilities are also updated.

The probability of internet lookup table at node i that has a radio link with node jis updated using the following equation:

$$p_{ij} = \frac{\tau_j \alpha_{ij} \beta_j}{\sum_j \tau_j \alpha_{ij} \beta_j}$$

where:

τ_j is the pheromone information updated by backwards ants

α_{ij} is the radio link power between nodes

β_j is the node power status

This approach does not take into account low power consumtion since this algorithm will be implemented at in-car entertaiment systems (ICE) and does not take into account memory limitations, this can be solved using a circular table in order to remove low probabilities. Real-time information is not need and backward information could also use a circular table when there are a lot of nodes.

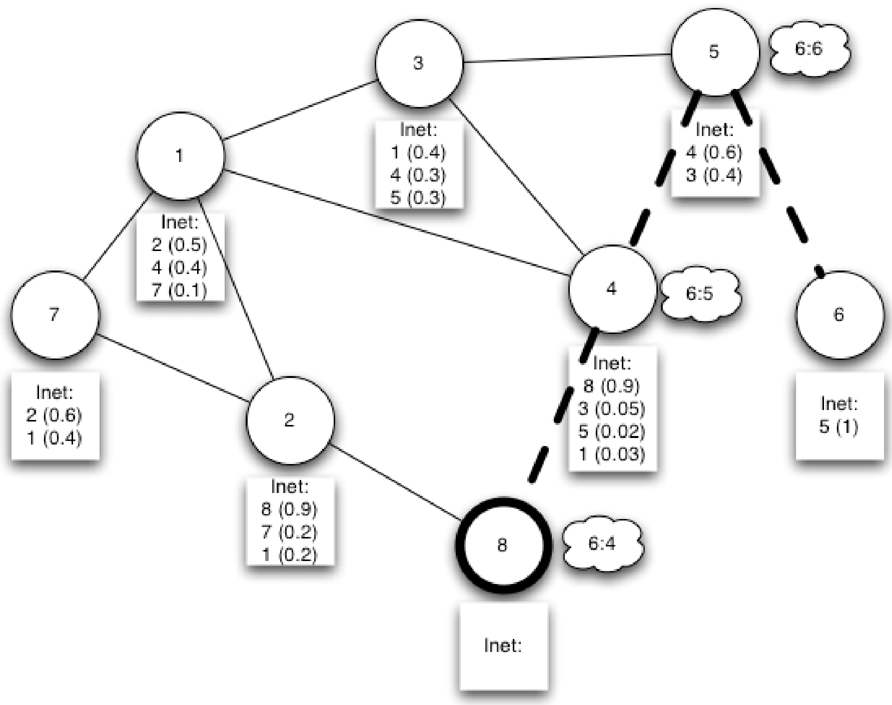


Figure 6.- WSN topology with internet routing table and backward information.

Future remarks

In-car entertaiment systems, sometimes referred to as ICE, is a collection of hardware devices installed into automobiles, or other forms of transportation, to provide audio and/or audio/visual entertainment, as well as automotive navigation systems (SatNav). This includes playing media such as CDs, DVDs, Freeview/TV, USB and/or other optional surround sound, or DSP systems. Also increasingly common in ICE installs are the incorporation of video game consoles into the vehicle.In Car Entertainment systems have been featured TV shows such as MTV's Pimp My Ride. In Car Entertainment has been become more widely available due to reduced costs of devices such as LCD screen/monitors, and the reducing cost to the consumer of the converging media playable technologies. Single hardware units are capable of playing CD, MP3, WMA, DVD.

MIT's CarTel project is investigating how cars themselves could be used as ubiquitous, highly reliable mobile sensors. At the Association for Computing Machinery's sixth annual Workshop on Foundations of Mobile Computing, members of the CarTel team presented a new algorithm that would optimize the dissemination of data through a network of cars with wireless connections. Researchers at Ford are already testing the new algorithm for possible inclusion in future versions of Sync, the in-car communications and entertainment system developed by Ford and Microsoft.

Described algorithms could be embedded into ICE in order to improve the routing algorithm since there is no need of real-time information retreival and, in some cases, no need of low power consumption.

Bibliography

[1] TinyOs web page. http://www.tinyos.net

[2] Viptos web page. http://ptolemy.eecs.berkeley.edu/viptos/

[3] Ptolemy II web page. http://ptolemy.eecs.berkeley.edu/ptolemyII/

[4] Elaine Cheong, Edward A. Lee, Yang Zhao: Viptos: a graphical development and simulation environment for tinyOS-based wireless sensor networks. SenSys 2005:302

[5] Christopher X. Brooks, Edward A. Lee, Stavros Tripakis: Exploring models of computation with ptolemy II. CODES ISSS 2010:331-332

[6] Yang Zhao, Yuhong Xiong, Edward A. Lee, Xiaojun Liu, Lizhi C. Zhong: The design and application of structured types in Ptolemy II. Int. J. Intell. Syst. (IJIS) 25(2):118-136 (2010)

[7] Kyungmin Bae, Peter Csaba Ölveczky: Extending the Real-Time Maude Semantics of Ptolemy to Hierarchical DE Models RTRTS 2010:46-66

[8] Martin Schoeberl, Christopher X. Brooks, Edward A. Lee: Code Generation for Embedded Java with Ptolemy. SEUS 2010:155-166

[9] Rajkumar K. Raval, Carlos Fernández, Chris J. Bleakley: Low-power TinyOS tuned processor platform for wireless sensor network motes. ACM Trans. Design Autom. Electr. Syst. (TODAES) 15(3) (2010)

[10] A. Sivagami, K. Pavai, D. Sridharan, S. A. V. Satya Murty: Energy and Link Quality Based Routing for Data Gathering Tree in Wireless Sensor Networks Under TINYOS - 2.X CoRR abs/1005.1739 (2010)

[11] Swarup Kumar Mitra, Ayon Chakraborty, Subhajit Mandal, Mrinal K. Naskar: Simulation of Wireless Sensor Networks Using TinyOS- A case Study CoRR abs/1004.4154 (2010)

[12] Doina Bucur, Marta Z. Kwiatkowska: Software verification for TinyOS. IPSN 2010:400-40151EESimon Kellner: Flexible Online Energy Accounting in TinyOS. REALWSN 2010:62-73

[13] Muhammad Hamad Alizai, Bernhard Kirchen, Jó Ágila Bitsch Link, Hanno Wirtz, Klaus Wehrle: TinyOS meets wireless mesh networks. SenSys 2010:429-430

[14] Nelson I. Dopico, Carlos Gil-Soriano, Inigo Arrazola, Santiago Zazo: Analysis of IEEE 802.15.4 Throughput in Beaconless Mode on micaZ under TinyOS 2. VTC Fall 2010:1-5

[15] Werner Backes, Jared Cordasco: MoteAODV - An AODV Implementation for TinyOS 2.0. WISTP 2010:154-169

[16] Angelo Paolo Castellani, Paolo Casari, Michele Zorzi: TinyNET - a tiny network framework for TinyOS: description, implementation, and experimentation. Wireless Communications and Mobile Computing (WICOMM) 10(1):101-114 (2010)

[17] Ajay C Solai Jawahar: Ant colony optimization for mobile ad-hoc networks. ECE-572 - Parallel and Distributed Computing. Rutgers University.

[18] Bandyopadhyay, M. andBhaumik, P.:Zone Based Ant Colony Routing In Mobile Ad-hoc Network. Communication Systems and Networks (COMSNETS), 2010, ISBN: 978-1-4244-5487-7, pp.: 1-10. (2010).

[19] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.

[20] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[21] K.Saleem, N.Fisal, S.Hafizah, S.Kamilah and Rozeha A. Rashid; Ant based Self-organized Routing Protocol for Wireless Sensor Networks . International Journal of Communication Networks and Information Security (IJCNIS), Vol. 1 (2). 2009.

Acknowledgment

This work has been partially supported by the Spanish Research Projects:

TRA2010-15645. "COMUNICACIONES EN MALLA PARA VEHICULOS E INFRAESTRUCTURAS INTELIGENTES" (Mesh communication with intelligent vehicles). (2010)

TEC2010-21303-C04-02. "ESTRUCTURAS RESONANTES PARA APLICACIONES DE SEÑAL FOTONICA DE BANDA ANCHA ". (2010).

Authors' Information

Nuria Gómez Blas *– Associate professor U.P.M Crtra Valencia km 7, Madrid-28031, Spain; e-mail: ngomez@eui.upm.es Research: DNA computing, Membrane computing, Education on Applied Mathematics and Informatics*

Luis F. de Mingo *– Associate professor U.P.M Crtra Valencia km 7, Madrid-28031, Spain; e-mail: lfmingo@eui.upm.es Research: Artificial Intelligence, Social Intelligence, Education on Applied Mathematics and Informatics*

Levon Aslanyan*– Head of Department, Institute for Informatics and Automation Problems, NAS RA, P.Sevak St. 1, Yerevan 14, Armenia, e-mail: lasl@.sci.am*

Vladimir Ryazanov*– Russian Academy of Science, Russian Federation; e-mail: rvvccas@mail.ru*