Simulation Tools in Wireless Sensor Networks: Ant Colony Optimization of a Local Routing Algorithm

Nuria Gómez Blas, Luis Fernando de Mingo López, Alberto Arteta Albert

***Abstract*—** Wireless Sensor 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. Network Management of such a sensor network is a very big challenge. Also the fast changing nature and the adhoc necessity of the network prevents the choice of a Centralized Solution which can decide the best route to route packets and at the same time minimize the different parameters like congestion, load, etc. Also no single node can take up the job of centralized manager due to the limited energy and processing capabilities of mobile nodes. Hence this has resulted in the need for a distributed approach which involves limited processing and power from the individual nodes but which work towards a concerted goal of routing and network management. This paper proposes a routing algorithm based on ant colonies. 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. A testing environment has been established for a future simulation.

***Keywords*—** Distributed Computing, Swarm Computing, Wireless Sensors Networks, Particle Swarm Optimization.

I. PRELIMINARIES

Distributed time-varying problems are the next big field to move to for researchers working on swarm-based optimization and problem-solving. The problem to be solved by any routing algorithm is to direct traffic from sources to destinations maximizing network performance while minimizing costs. There are many possible routing problems differing mainly by the characteristics of the network and of the traffic. In real networks traffic conditions are constantly changing, and the structure of the network itself may fluctuate. Because there are usually many possible pathways for one message to go from a given node to another node, it is possible to make routing algorithms adaptive enough to overcome local congestion. Static routing, whereby routing remains fixed independent of the current states of the network and user traffic, is therefore almost never implemented: most routing schemes respond in some way to changes in network or user traffic states. But there exists a wide spectrum of dynamic routing systems, which vary dramatically in their speed of response and in the types of changes they respond to [1]. Dynamic routing requires more computational resources than static or quasi static routing. It relies on active participation of entities within the network to measure user traffic, network state and performance, and to compute routes.

Schoonderwoerd et al. [2] have proposed a very interesting adaptive routing algorithm based on the use of many simple agents, called ants, that modify the routing policy at every node in a telephone network by depositing a virtual pheromone trail on routing tables entries. The ants' goal is to build, and adapt to load changes at run time, routing tables so that network performance is maximized. The network performance is measured by the rate of incoming calls which are accepted (incoming calls can be either accepted or rejected depending on the network resources available at call set-up time).

Di Caro and Dorigo [3], [4], [5] have introduced an adaptive routing algorithm based on ant colonies that explore the network with the goal of building routing tables and keeping them adapted to traffic conditions. Their algorithm, called *AntNet*, although based on principles similar to those of Schoonderwoerd et al.'s approach [2], has some important differences: *AntNet* can be applied to both connection-oriented and connectionless types of communication networks, ants collect information which is used to build local parametric models of the network status used to compute reinforcements to change the probabilistic routing tables and *AntNet* has been tested on a packet-switching network model and its performance has been compared to the performance of well-known existing routing algorithms.

AntNet, as well as most of the other ACO routing algorithms designed after AntNet, exhibits a number of interesting properties: it works in a fully distributed way, is highly adaptive to network and traffic changes, uses lightweight mobile agents (called ants) for active path sampling, is robust to agent failures, provides multipath routing, and automatically takes care of data load spreading.

AntNet's performance has been extensively tested in simulation, considering different networks and traffic patterns, and compared to several state-of-the-art routing algorithms. In the great majority of the considered situations, AntNet has largely outperformed all its competitors, showing excellent adaptivity and robustness. AntNet has been also tested in small physical networks, confirming the good performance also in these real-world tests.

Manuscript received May 2, 2012: Revised version received May XX, 2012. This work was supported in part by the Spanish Ministry of Education under Grants TEC2010-21303-C04-02 and TRA2010-15645.

Nuria Gómez is with Dpto. Organizacón y Estructura de la Información, Escuela Universitaria de Informática, Universitad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain, ngomez@eui.upm.es

L.F. de Mingo is with Dept. Organizacón y Estructura de la Información, Escuela Universitaria de Informática, Universitad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain, lfmingo@eui.upm.es

Alberto Arteta is with Dpto. Matemática Aplicada, Escuela Universitaria de Informática, Universitad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain, aarteta@eui.upm.es

II. Wireless sensor network (WSN)

A wireless sensor network is composed of physically disconnected sensors that must communicate wirelessly to collect data and coordinate activity [6], [7]. The nature and exact speciications of the network can vary depending upon the application. The traditional WSN model, originally focused on military applications, involved a random distribution of sensors throughout the target area, in addition to one or more sink nodes. Since the network is isolated, the sensors have limited sensing and communication capabilities as well as limited battery life, unless fitted with energy-generation technology (e.g., solar panels). Transmissions are multi-hop, passing along the most cost-effective path in the network graph from the originating sensor to the sink. Sink nodes are typically more powerful then the ordinary sensors and may have greater energy capacity and better sensing and communicating equipment; they may even be physically connected to other sink nodes in the network. Sensors may be stationary or mobile, allowing for dynamic configurations and layouts of the network depending on certain events or commands. Applications of WSNs include target detection, tracking, pursuit, and passive area monitoring for security purposes. In recent WSN developments, networks can be heterogeneous, incorporating sensors with different capabilities and resources. The environment and surrounding topology can also affect how the network operates: sensors at higher elevations may be able to communicate further, while those in valleys or surrounded by obstacles may have limited sensing range. The motion of a sensor may be outside its control, as in the case of cell phones passing in and out of range of various cell towers or cars travelling along a highway. These adhoc networks must be robust to sudden changes and must be able to quickly coordinate sensor communications to preserve quality of service.

III. Simulation of wireless sensor networks: software tools

Simulation is essential to study WSN, being the common way to test new applications and protocols in the field. This fact has brought a recent boom of simulation tools available to model WSN. However, obtaining reliable conclusions from research based on simulation is not a trivial task. There are two key aspects that should be evaluated before conducting experiments: (1) The correctness of the model and (2) the suitability of a particular tool to implement the model.

Implementing a complete model requires a considerable effort. A tool that helps to build a model is needed, and the user faces the task of selecting the appropriate one. Simulation software commonly provides a framework to model and reproduce the behavior of real systems. However, actual implementation and secondary goals of each tool differ considerably, that is, some may be designed to achieve good performance and others to provide a simple and friendly graphical interface or emulation capabilities.

The aim of this paper is to provide some insight on the building blocks of a general simulation model for WSN, introducing its specific issues. Also, to facilitate newcomers the selection of the most appropriate tool for their needs, the most extended WSN simulation environments are reviewed.

A. TinyOS

TinyOS [8], [9] 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 multi-hop, network-wide sub-millisecond time synchronization through the Flooding Time Synchronization Protocol, developed by researchers at Vanderbilt University.

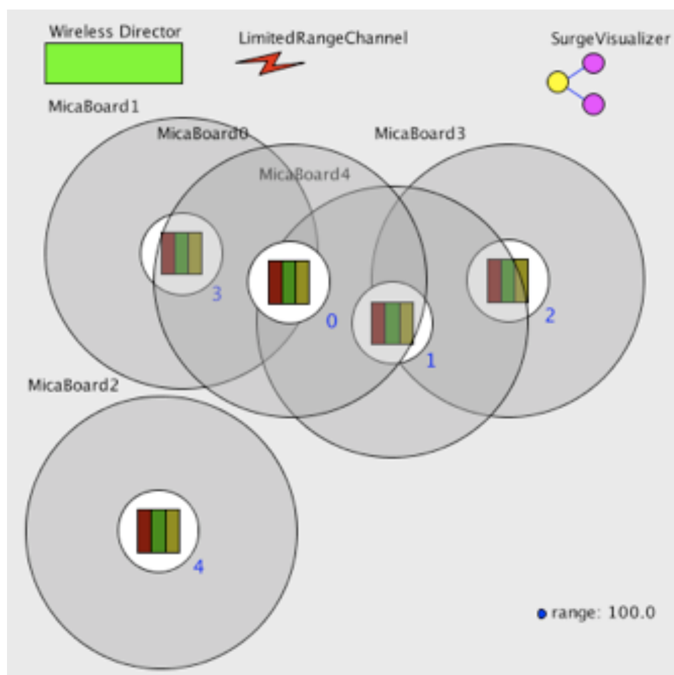


Fig. 1. High level definition of sensors on space.

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

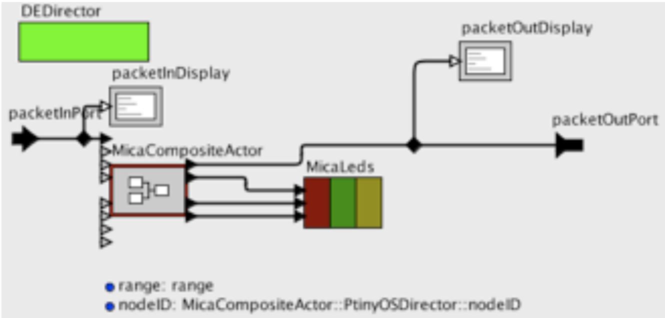


Fig. 2. Sensor configuration, signals and leds.

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. These three protocols represent a gradual evolution towards more efficient algorithms. Generally speaking, applications should use DHV.

TinyOS includes support for reprogramming a multi-hop 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.

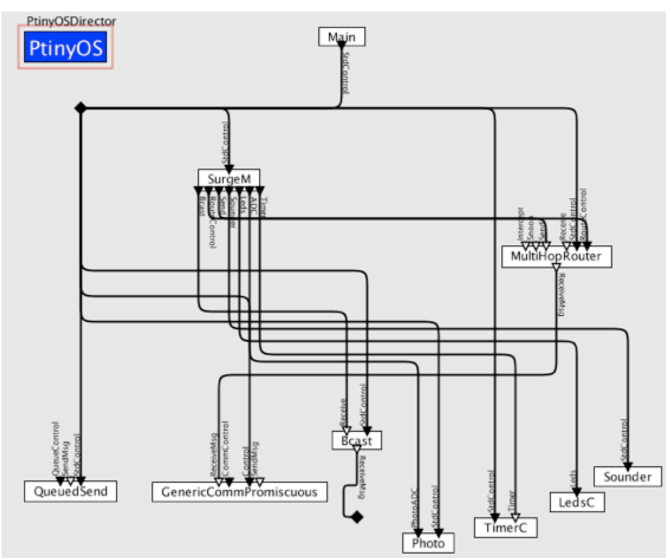


Fig. 3. Low level programming of a given sensor.

B. Viptos and PTolemyII

Viptos (Visual Ptolemy and TinyOS) [10], [11] is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. Viptos allows developers to create block and arrow diagrams, see figures 1, 2 and 3; to construct TinyOS [8], [9] 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 [10], [11] and Ptolemy II [12], [13]. 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 [13] 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 [11], 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.

TinyViz [8], [9] is a Java-based GUI that allows you to visualize and control the simulation as it runs, see figure 4, 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.

IV. Natural Computing

Natural sciences, and especially biology, represented a rich source of modeling paradigms. Well-defined areas of artificial intelligence (genetic algorithms, neural networks), mathematics, and theoretical computer science (L systems, DNA computing) are massively influenced by the behavior of various biological entities and phenomena. In the last decades or so, new emerging fields of so-called natural computing identify new (unconventional) computational paradigms in different forms. There are attempts to define and investigate new mathematical or theoretical models inspired by nature, as well as investigations into defining programming paradigms

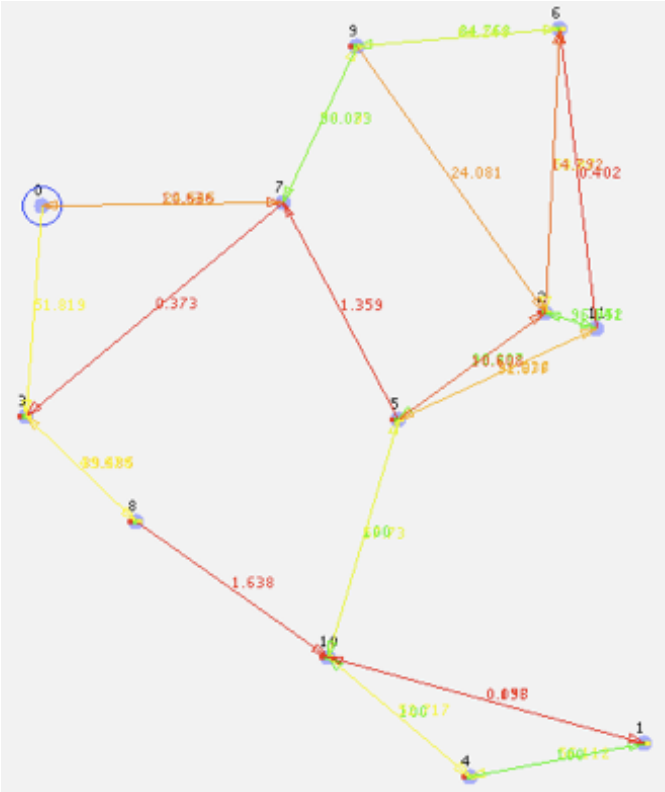


Fig. 4. TinyViz visualization of a sensor network.

that implement computational approaches suggested by biochemical phenomena. Especially since Adleman's experiment, these investigations received a new perspective. One hopes that global system-level behavior may be translated into interactions of a myriad of components with simple behavior and limited computing and communication capabilities that are able to express and solve, via various optimizations, complex problems otherwise hard to approach.

A number of computational paradigms, inspired or gleaned from biochemical phenomena, are becoming of growing interest building a wealth of models, called generically Molecular Computing. New advances in, on the one hand, molecular and theoretical biology, and on the other hand, mathematical and computational sciences promise to make it possible in the near future to have accurate systemic models of complex biological phenomena.

Natural computation [14], also called natural computing, is the field of research that works with computational techniques inspired in part by nature and natural systems. The aim of such research is to develop new computational tools (in software, hardware or wet-ware) for solving complex, usually conventionally-hard problems. This often leads to the synthesis of natural patterns, behaviors and organisms, and may result in the design of novel computing systems that use natural media with which to compute. Natural computing can be divided into three main branches:

- Computing inspired by nature (also called biologically inspired computing): This makes use of nature as inspiration for the development of problem solving techniques. The main idea of this branch is to develop computational tools (algorithms) by taking inspiration from nature for the solution of complex problems;
- The simulation and emulation of nature by means of computing: This is basically a synthetic process aimed at creating patterns, forms, behaviors, and organisms that (do not necessarily) resemble life-as-we-know-it. Its products can be used to mimic various natural phenomena, thus increasing our understanding of nature and insights about computer models; and
- Computing with natural materials: This corresponds to the use of natural materials to perform computation, thus constituting a true novel computing paradigm that comes to substitute or supplement the current silicon-based computers.

Two popular variants of swarm models exist, those inspired by studies of social insects such as ant colonies, and those inspired by studies of the flocking behavior of birds and fish.

Ant colony optimization (ACO) is a class of optimization algorithms modeled on the actions of an ant colony. ACO methods are useful in problems that need to find paths to goals. Artificial 'ants' - simulation agents - locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment, see algorithm **??**. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions [15], [16]. One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee.

Particle swarm optimization (PSO) is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication channel between the particles [17], [18]. Particles then move through the solution space, and are evaluated according to some fitness criterion after each time step. Over time, particles are accelerated towards those particles within their communication grouping which have better fitness values. The main advantage of such an approach over other global minimization strategies such as simulated annealing is that the large number of members that make up the particle swarm make the technique impressively resilient to the problem of local minima.

Stochastic Diffusion Search (SDS) is an agent-based probabilistic global search and optimization technique best suited to problems where the objective function can be decomposed into multiple independent partial-functions. Each agent maintains a hypothesis which is iteratively tested by evaluating a randomly selected partial objective function parameterised by the agent's current hypothesis. In the standard version of SDS such partial function evaluations are binary, resulting in each agent becoming active or inactive. Information on hypotheses is diffused across the population via inter-agent communication. Unlike the stigmergic communication used in ACO, in SDS agents communicate hypotheses via a one-to-one communication strategy analogous to the tandem running procedure observed in some species of ant. A positive feedback mechanism ensures that, over time, a population of agents stabilise around the global-best solution. SDS is both an efficient and robust search and optimization algorithm, which has been extensively mathematically described.

V. Self-organizing routing

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), see algorithm 2, and the quadratic assignment problem (QAP), and later introduced the ACO meta-heuristic.

Algorithm 1 ACO Algorithm

Initialize pheromone values
repeat
  for ant $k \in 1, \cdots, m$ **do**
    construct a solution
  end for
  for all pheromone values **do**
    decrease the value by a certain percentage evaporation
  end for
  for all pheromone values corresponding to good solutions **do**
    increase the value intensification
  end for
until stopping criterion is met

Algorithm 2 ACO-TSP Algorithm

Initialize pheromone values
repeat
  for ant $k \in 1, \cdots ., m$ **do**
    $S = 1, \cdots, n$
    choose city i with probability p_{0i}
    repeat
      choose city $j \in S$ with probability p_{ij}
      $S = Sj$
      $i = j$
    until $S = \emptyset$
  end for
  for all i, j **do**
    $\tau ij = (1\rho)\tau_{ij}$
  end for
  for all i, j in iteration best solution **do**
    $\tau_{ij} = \tau_{ij} + \Delta$
  end for
until stopping criterion is met

Hence the primary goal in a mobile network [19] is to efficiently establish one or more routes between two nodes so that they can communicate reliably. Such a network is characterized by the following challenges.

- The network topology can change dynamically due to the random movement of nodes.
- Also any node may leave/join the network and the protocol must adapt accordingly.
- Although no guarantee of service can be provided, the protocol must be able to maximize the reliability of packet in the network for the given conditions.

There are two types of ants applied in the algorithm, see figure 6, 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 [20]. 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.

POSANT Routing Algorithm [21], [22] 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. Ant Colony Routing Algorithm with Zones [23]. 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 mobile ad-hoc network (MANET) is a collection of mobile nodes which communicate over radio [19].

The following set of core properties characterizes ACO instances for routing problems:

- provide traffic-adaptive and multipath routing,
- rely on both passive and active information monitoring and gathering,
- make use of stochastic components,
- do not allow local estimates to have global impact,
- set up paths in a less selfish way than in pure shortest path schemes favoring load balancing,
- show limited sensitivity to parameter settings.

A. *Local routing*

Ants are mobile agents that migrate from one node to an adjacent one searching for feasible paths between source and destination nodes. ACOs solution components (and phantasmata) correspond to network nodes, and, accordingly, routing tables correspond to pheromone tables T^k in which each

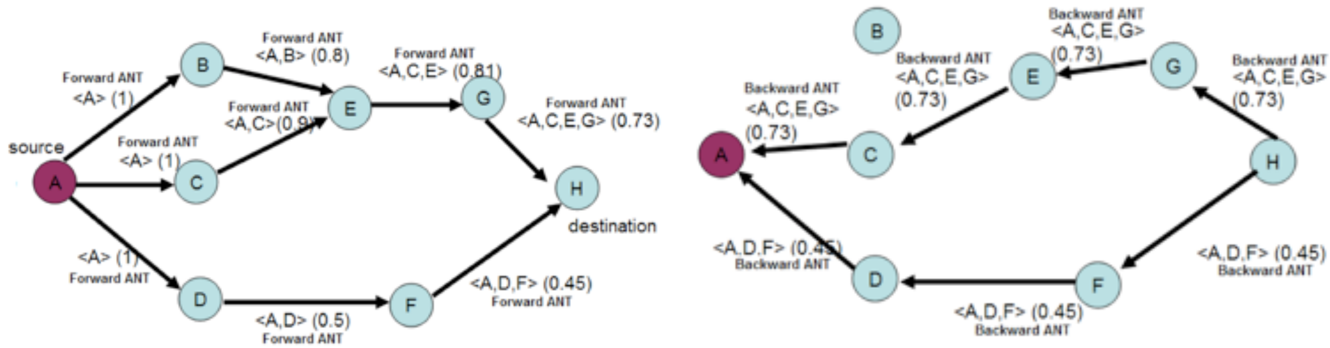


Fig. 5. Ant colony routing in mobile ad-hoc networks (MANET) protocol [19].

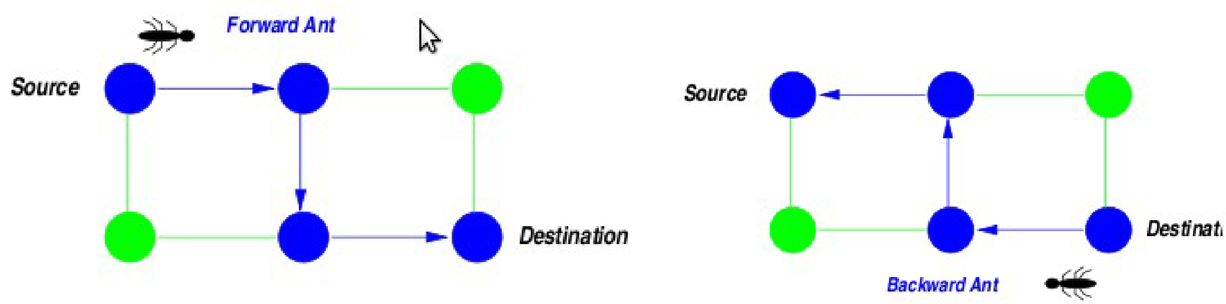


Fig. 6. Ant colony routing: forward and backward ants behavior.

pheromone variable τ_{nd}^{k} holds the estimated goodness of selecting ks neighbor n to forward a data packet toward d.

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 neighborhood nodes/motes that have internet connection, see table I. 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 j is updated using the following equation:

$$p_{ij} = \frac{\tau_j \alpha_{ij} \beta_j}{\sum_j \tau_j \alpha_{ij} \beta_j} \quad (1)$$

TABLE I
LOCAL ROUTING TABLE STORE IN EACH SENSOR

Internet Lookup Table (Mote ID: Probability)	Data Packet Information (Source Mote, Route)
$7 : 0.9$	$(2, 3)$
$5 : 0.6$	$(2, 5)$
$3 : 0.3$	$(2, 1)$
$1 : 0.4$	$(2, 6)$

, where τ_j is the pheromone information updated by backwards ants, α_{ij} is the radio link power between nodes and β_j is the node power status.

This approach does not take into account low power consumption [24], [25] since this algorithm will be implemented at in-car entertainment 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.

VI. Testing Evironment

In the experiments, the following networks have been considered: , two networks modeled on the characteristics of two different real-world networks, one network with somehow regular grid-like topology, and two classes of randomly generated networks with a rather high number of nodes. Their characteristics are described in the following of this subsection. For each network a triple of numbers (μ, σ, N) is given, indicating respectively the mean shortest path distance in terms of hops between all pairs of nodes, the variance of this mean value, and the total number of nodes. These three numbers are intended to provide a measure concerning the degree of connectivity and balancing of the network. It can be in general said that the difficulty of the routing problem, for the same input traffic, increases with the value of these numbers. The internet sink is place randomly on a node and changes at regular time intervals.

- NSFNET (2.2, 0.8, 14) is the old USA T1 backbone (1987). NSFNET is a WAN composed of 14 nodes and 21 bi-directional links with a bandwidth of 1.5Mbit/s. Its topology is shown in figure 7(a). Propagation delays range from 4 to 20 msec. NSFNET is a well balanced network.
- NTTnet (6.5, 3.8, 57) is modeled on the former NTT (Nippon Telephone and Telegraph company) fiber-optic corporate backbone. NTTnet is a 57 nodes, 162 bi-directional links network. Link bandwidth is of 6 Mbit/sec, while propagation delays range from around 1 to 5 msec. The topology is shown in figure 7(b). NTTnet is not a well balanced network.
- 6x6Net (6.3, 3.2, 36) is a 36 nodes network with a regular topology and a sort of bottleneck path separating the two equal halves of the network. This network has been introduced by Boyan and Littman [26] in their work on Q-Routing. In a sense, this is a pathological network, considered its regularity and the bottleneck path. All the links have bandwidth of 10 Mbit/s and propagation delay of 1 msec., see figure 7(c).
- Random Networks (4.7, 1.8, 100) and (5.5, 2.1, 150) are two sets of randomly generated networks of respectively 100 and 150 nodes. The level of connectivity of each node has been forced to range between 2 and 5. The reported values for the mean shortest path distances and their variances are averages over the 10 randomly generated networks that have been used for the experiments. Every bi-directional link has the same bandwidth of 1Mbit/sec, while the propagation delays have been generated in a uniform random way over the interval [0.01, 0.001].

Previous experiments will be run in a TinyOS environment with the local routing algorithm proposed along this paper.

VII. Conclusions and Future remarks

Sensor Ad-hoc Networks provide a wide array of challenges in routing and network management due to their dynamic and distributed nature and various protocols have been studied and implemented in view of these needs and challenges. In this paper, an alternate approach inspired by concepts of emergence and self-organization in biological systems, has been discussed and implemented. The approach tries to optimize routing by finding the best shortest path,

This algorithm has a lot of scope for future improvements. Though the algorithm has better control over congestion and load, its adaptability to dynamic node movements is not very different from existing algorithms. Hence future work lies in incorporating factors like signal strength into the route metrics so as to predict link breaks before they actually occur and redirect to other routes. Also extensive testing needs to be performed using the current algorithm under varying traffic and movement scenarios.

In-car entertainment 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, Free view/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.

MITs CarTel project is investigating how cars themselves could be used as ubiquitous, highly reliable mobile sensors. At the Association for Computing Machinerys 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 retrieval and, in some cases, no need of low power consumption.

References

[1] M. Steenstrup, *Routing in Communications Networks*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[2] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive Behaviour*, vol. 5, pp. 169–207, 1996.

[3] G. D. Caro and M. Dorigo, "Antnet: A mobile agents approach to adaptive routing," Universite Libre de Bruxelles, IRIDIA/97-12, 1997.

[4] ——, "A study of distributed stigmergetic control for packet-switched communitcations networks," Universite Libre de Bruxelles, Tech. Rep. IRIDIA/97-20, 1997.

[5] ——, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Resources*, vol. 9, pp. 317–365, 1998.

[6] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, pp. 54–61, 2004.

[7] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using an ant colony optimization (aco) router chip," *Sensors*, vol. 9, pp. 909–921, 2009.

[8] TinyOS, "http://www.tinyos.net."

[9] M. H. Alizai, B. Kirchen, J. A. B. Link, H. Wirtz, and K. Wehrle, "Tinyos meets wireless mesh networks," *SenSys*, pp. 429–430, 2010.

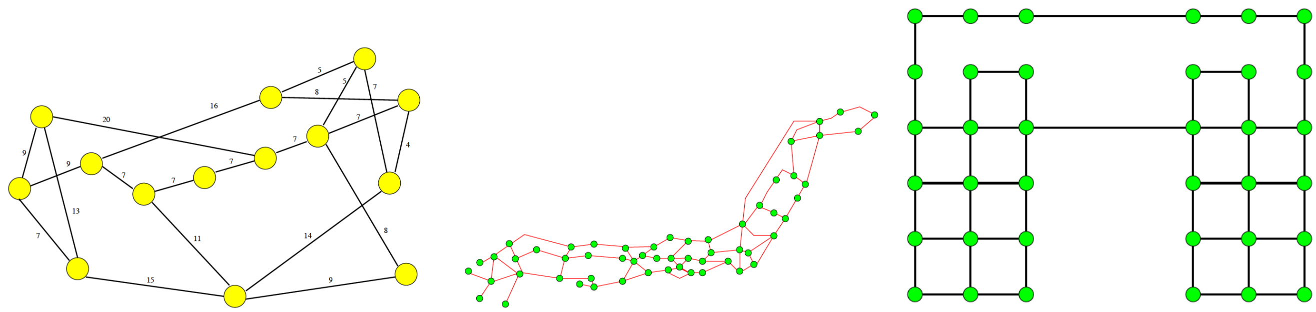
[10] ViptOS, "http://ptolemy.eecs.berkeley.edu/viptos/."

[11] E. Cheong, E. A. Lee, and Y. Zhao, "Viptos: a graphical development and simulation environment for tinyos-based wireless sensor networks," *SenSys*, p. 302, 2005.

[12] PtolemyII, "http://ptolemy.eecs.berkeley.edu/ptolemyii/."

[13] C. X. Brooks, E. A. Lee, and S. Tripakis, "Exploring models of computation with ptolemyii," *Codes*, pp. 331–332, 2010.

[14] Wikipedia, "http://en.wikipedia.org."



(a) NSFNET. Each edge in the graph represents a pair of directed links. Link bandwidth is 1.5 Mbit/sec, propagation delays range from 4 to 20 msec and are indicated by the numbers reported close to the links.

(b) NTTnet. Each edge in the graph represents a pair of directed links. Link bandwidth is 6 Mbit/sec, propagation delays range from 1 to 5 msec.

(c) 6x6Net. Each edge in the graph represents a pair of directed links. For all the links the bandwidth is equal to 10 Mbit/sec and the propagation delay is equal to 1 msec.

Fig. 7. MANET different testing topologies.

[15] D. S. Kumar and R. S. Bhuvaneswaran, "Alrp: Scalability study of ant based local repair routing protocol for mobile adhoc networks," *WSEAS TRANSACTIONS on COMPUTER RESEARCH*, vol. 3, no. 4, pp. 225–233, 2008.

[16] R. Jovanovic, M. Tuba, and D. Simian, "Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem," *WSEAS TRANSACTIONS on COMPUTERS*, vol. 9, no. 1, pp. 83–92, 2010.

[17] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. San Mateo, California: Morgan Kauff-man, 2001.

[18] *Swarm Intelligence: From Natural to Artificial Systems*. Oxford: Oxford University Press, 1999.

[19] A. C. S. Jawahar, "Ant colony optimization for mobile ad-hoc networks," *ECE*, vol. 572.

[20] K. Saleem, N. Fiscal, S. Hafizah, S. Kamilah, and A. Rozeha, "Ant based self-organized routing protocol for wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, 2009.

[21] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, 1992.

[22] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Conference Europeene sur la Vie Artificielle*, 1991, pp. 132–142.

[23] M. Bandyopadhyay and P. Bhaumik, "Zone based ant colony routing in mobile ad-hoc network," in *Communication Systems and Networks (COMSNETS)*, 2010, pp. 1–10.

[24] R. Saravanakumar, S. G. Susila, and J. Raja, "Energy efficient constant cluster node scheduling protocol for wireless sensor networks," *WSEAS TRANSACTIONS on COMMUNICATIONS*, vol. 10, no. 1, pp. 122–128, 2011.

[25] A. Balamurugan and T. Purusothaman, "Ipsd: New coverage preserving and connectivity maintenance scheme for improving lifetime of wireless sensor networks," *WSEAS TRANSACTIONS on COMMUNICATIONS*, vol. 11, no. 1, pp. 26–36, 2012.

[26] J. A. Boyan and M. L. Littman, "Packet routing in dinamically changing networks: A reinforcement learning approach," *Advances in Neural Information Processing Systems*, vol. 6, pp. 671–678, 1994.

Luis Fernando de Mingo López is currently an associate professor in the technical University of Madrid from 2005. He works in the Department Organization of data structures. His main area of interest includes the study of new computational models such as neural networks, ant colony optimization and particle swarm optimization. He holds his Ph.D. in this field and has published more than 30 papers. He is also an editor of the International Scientific Society Journal.

Nuria Gómez Blas is currently an associate professor in the technical University of Madrid from 2005. She works in the Department Organization of data structures. Her main area of interest includes the study of new computational models such as neural networks and particle swarm optimization, and the innovation of educational method optimization. She holds her Ph.D. in this field. Ph.D. in this field and has published papers in prestigious journals and in symposium including Engineering Application of Artificial Intelligence, Journal of Computational Sciences, etc. She has a BsC on Biology and other on Computer Science.

Alberto Arteta Albert is currently an associate professor of the technical University of Madrid from 2007. He works in the Department of Applied mathematics. His main area of interest includes the study of new computational models such as membrane computing, cells computing, evolutionary computing and neural networks. He holds his Ph.D. in this field and has published papers in prestigious journals and in symposium including Engineering Application of Artificial Intelligence, Evolutionary computation, and IEEE membrane computing symposium. Furthermore, in the last 13 years, he had been working mainly in Holland and Spain for private companies. Most of his roles have been as a developer, system engineer and database consultant. The last companies he has worked are IBM, CITCO, ROCHE FARMA, PROGRESS, MCAFEE, AMADEUS IT and EVERIS SPAIN.