ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 2, Issue 8, August 2013*

# Power Aware Routing Protocol for MANET's using Swarm Intelligence

Sridhar H.S [1], Dr. M. Siddappa [2], Dr.G.C.Bhanu Prakash[3]

Assistant Professor, Dept. of Electrical and Electronics Engg , Siddaganga Institute of Technology, Tumkur, India[1]

Professor and Head of the Department, Dept. of Computer science and Engg, Sri Siddhartha Institute of Technology, Tumkur, India[2]

Professor, Dept. of Computer science and Engg, SIR M.Visvesvaraya Institut of Technology, Bangalore, India [3]

**Abstract**: Wireless networking has witnessed an explosion of interest from consumers in recent years for its applications in mobile and personal communications. As wireless networks become an integral component of the modern communication infrastructure, energy efficiency will be an important design consideration due to the limited battery life of mobile terminals. As MANET's are generally battery-powered devices, the critical aspects to face concern how to reduce the energy consumption of nodes, so that the network lifetime can be extended to reasonable times. Since the network interface is a significant consumer of power, considerable research has been devoted to low-power design of the entire network protocol stack of wireless networks in an effort to enhance energy efficiency. This paper we presents that insect colonies based intelligence – commonly referred to as Swarm Intelligence (SI) – provides an ideal metaphor for developing routing protocols for MANETs. In this context, we propose a new routing protocol for MANETs – SensorBee – inspired by the foraging principles of honey bees.

**Keywords**: Energy efficiency, Power management, Swarm Intelligence

## I. Introduction

An ad hoc network is the cooperative engagement of a collection of mobile nodes without the required intervention of any centralized access point or existing infrastructure. There is an increasing trend to adopt ad hoc networking for commercial uses; however, their main applications lie in military, tactical and other security-sensitive operations. In these and other applications of AD-HOC networking, energy efficient secure routing is an important issue [3].Wireless hosts are usually powered by batteries which provide a limited amount of energy. Therefore, techniques to reduce energy consumption are of interest. One way to conserve energy is to use power saving mechanisms. Power saving mechanisms allows a node to enter a snooze state by powering off its wireless network interface when deemed reasonable. Another alternative is to use power control schemes which suitably vary transmit power to re- duce energy consumption .In addition to providing energy saving, power control cans potentially be used to improve spatial reuse of the wireless channel [1].

Routing in MANETs has been a challenging task primarily because of limited hardware resources available at a sensor node. Consequently, routing protocols are designed with low processing complexity and minimum communication overhead. Since the sensor nodes mostly operate in pervasive environments with no user intervention; therefore, routing must be done through distributed and decentralized controllers at each node, which through local and partial information should be able to self-organize and take adaptive routing decisions in response to changing external environments. Moreover, the protocols must be scalable, performance efficient with an ability to keep the network alive for a longer period of time [6].

Swarm intelligence (SI) [7] is a relatively novel field that was originally defined as "Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insects and other animal societies" [7]. However, nowadays it refers more generally to the study of the collective behavior of systems composed of many components that coordinate using decentralized controls and self-organization.

Agents in a bee colony – although have limited individual capabilities through local coordination:

- produce an highly organized and efficient system level behavior,

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 2, Issue 8, August 2013*

- show adaptively to a constantly changing environment
- exhibit resilience to loss of individuals
- scale well to larger populations.

Therefore, we believe that it can serve as an ideal metaphor for developing a routing protocol for MANETs which have identical characteristics. In this context, the objective is to engineer an event-driven, simple, scalable, reliable, decentralized and energy-efficient multipath routing protocol for WSNs through nature-inspired simple bee agents. The new routing protocol should meet the following set of requirements that are fundamentally important in MANETs.

- The bee agent model should be simple and easily realizable. Simplicity of a bee agent in directly requires that its size should be minimal.
- Routing protocol must be scalable to large network topologies and able to handle high traffic loads. Routing protocol must conserve energy in every possible manner without compromising the network performance in terms of packet delivery.

## II. CHALLENGES IN MOBILE AD-HOC NETWORKS

Ad-hoc networks have to suffer many challenges at the time of routing. Dynamically changing topology (due to Brownian motion of the nodes of the network) and no centralized infrastructure are the biggest challenges in the designing of an Ad-hoc network. The position of the nodes in an Ad-hoc network continuously varies due to which we can't say that any particular protocol will give the best performance in each and every case topology varies very frequently so we have to select a protocol which dynamically adapts the situation. Another challenge in MANET is limited bandwidth. If we compare it to the wired network then wireless network has less and more varying bandwidth.So, bandwidth efficiency is also a major concern in ad-hoc network routing protocol designing because sometimes data has to be transmitted within real time constraints. Limited power supply is the biggest challenge of an ad-hoc network so if we want to increase the network lifetime (duration of time when the first node of the network runs out of energy) as well the node lifetime then we must have an energy efficient protocol. So an ad-hoc routing protocol must meet all these challenges to give the average performance in every case[3]. The main challenges in mobile ad-hoc networks are as follows:

- Limited Power Supply
- Dynamically Changing Topology
- Limited Bandwidth
- Security
- Mobility-induced route changes
- Mobility-induced packet losses
- Battery constraints

## III. ENERGY EFFICIENT ROUTING

Energy is a limiting factor in case of Ad-hoc networks. Routing in ad-hoc networks has some unique characteristics.

- First- Energy of nodes is crucial and depends upon battery which has limited power supply.
- Second- Nodes can move in an uncontrolled manner so frequent route failures are possible.
- Third-Wireless channels have lower and more variable bandwidth compare to wired network.

Energy efficient routing protocols are the only solution to above situation. Most of the work of making protocols energy efficient has been done on "on demand routing protocols" because these protocols are more energy efficient rather than proactive protocols but still these have some drawbacks which have been discussed in the next section. Energy efficiency can also be achieved by sensible flooding at the route discovery process in reactive protocols. And energy efficiency can also be achieved by using efficient metric for route selection such as cost function, node energy, battery level etc. Here energy efficiency doesn't mean only the less power consumption here it means increasing the time duration in which any network maintains certain performance level. We can achieve the state of energy efficient routing b increasing the network lifetime and performance and all the protocols discussed in this paper are based on this concept [3][4][5]. MANETs contain large sets of resource constrained nodes. Therefore, design of effective, robust, and scalable routing protocols for WSNs is an extremely challenging task. In comparison, the domain of swarm intelligence offers algorithmic design principles, inspired by complex adaptive biological systems, which well match the constraints and the challenge of WSNs.

## IV. GENERAL FRAMEWORK FOR SI-BASED ROUTING

In this section, a unified view of SI-based algorithms a common modular framework is provided in which the different instances can be put in. It serves two purposes. First, it provides a common reference framework to describe and compare the different implementations of SI-based routing algorithms Second, it is also aimed to define a general architecture that can guide the design of future SI algorithms for network routing. In addition, the general engineering guidelines to design the components and the functioning of an SI router and define the behavior of the control agents used to setup routing paths are also provided.
The proposed framework consists of five top level modules and some additional sub modules. The ensemble of these modules and sub modules implements the architecture and the operations at the node router. The top level modules are: (i) Mobile agents generation and management,(ii) Routing information database (RID), (iii) Agent structures, (iv) Agent communications, and (v) Packet forwarding. Figure 1

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 2, Issue 8, August 2013*

summarizes the characteristics of the different modules and their relationships.

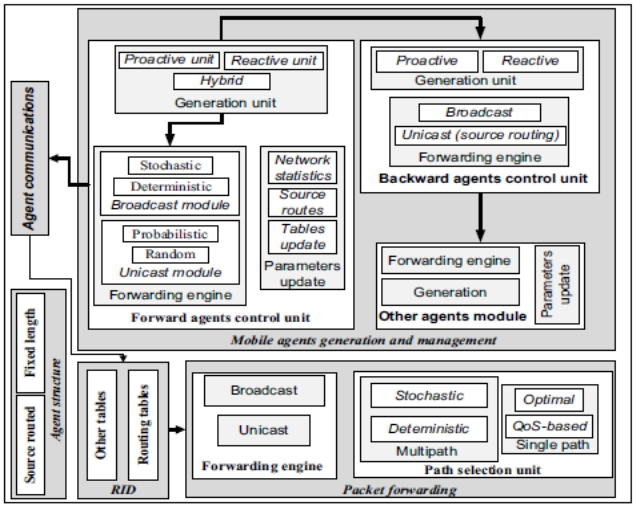


Fig 1: Diagram of the common routing framework for SI routing protocols

## V. SENSORBEE: ARCHITECTURE AND WORKING

SensorBee is an event driven multi-path routing protocol for MANETs  SensorBee discovers paths only when they are needed. The source node maintains all the discovered paths between a <source, sink> pair and routes different events stochastically on these multiple paths. The paths get a priority on the basis of a reward value which in turn is a function of the path length(hops) and the least energy level of a node on the path.

### V.I BEE AGENT MODEL

SensorBee works with four types of agents: packers, scouts, foragers and the swarms. Packers are classified as static agents because they perform their tasks within sensor node. Deferent mobile agents though have the same structure – consisting of header and payload fields but they undertake deferent types of tasks. A brief description of each type of agent is as follows.

### V.I.1SWARMING

Foragers are implicitly piggy-backed in the lower link acknowledgement packets to the source node to save energy. However, sometimes they need to be explicitly transported back to their source nodes. A swarm agent exactly serves this purpose. Foragers wait for a certain amount of time at the sink node and then take the initiative to build a swarm of waiting foragers. A swarm can transport multiple foragers in its payload back to the source node. A swarm like foragers is also routed on the reverse links.  The foragers should have a return path to get back to the source node from the sink node. Otherwise, the source node runs out of foragers and subsequently loses path to the destination. It is already mentioned that the foragers are either implicitly piggy-backed in the lower layer acknowledgement packets or swarms are used to explicitly transport them back to the source node. A swarm encapsulates all foragers belonging to its own group – same path ID foragers –in its payload. The swarm is then routed towards the source node using the reverse link entries (previous hop) in the forwarding tables. In addition to this, a swarm does not advertise a path if its minimum remaining energy level is below certain threshold, say pL, provided that better quality paths are available. Consequently, the poor quality paths are gradually removed from the routing tables.

### PACKERS

Packers behave like the food-storer bees in a hive. Their major responsibility is to receive packets coming from the upper layer and locate an appropriate forager (route) for them. Once a forager is found, packet is encapsulated in its payload and the packer starts waiting for the next packet. Failure in locating a forager is an indication to the packer that no route exists for the sink.

### SCOUTS

Like their natural counterparts, scouts explore the network in search of a potential sink node. Scouts are classified into two categories: forward scouts and backward scouts. A scout is uniquely identified by its agent ID and the source node ID. Forward scouts propagate in the network using the broadcasting principle. During the exploration of the network, they do not construct a source routing header. As a result, their size becomes independent of the path length that helps SensorBee to scale to large networks. Once forward scout reaches a sink node, it delivers the event to the upper layer and starts its return journey as a backward scout. Its task is to build a path leading from the sink to the source node and report the quality of the discovered path once it reaches the source node.

### SCOUTING

The scouting is divided into two steps: forward scouting and backward scouting. Forward scouting is initiated when a path to a sink node is not available. Forward scouts explore the network and look for a potential sink node. Once a sink node is found, the backward scouts establish multiple paths between the <source, sink> pair.

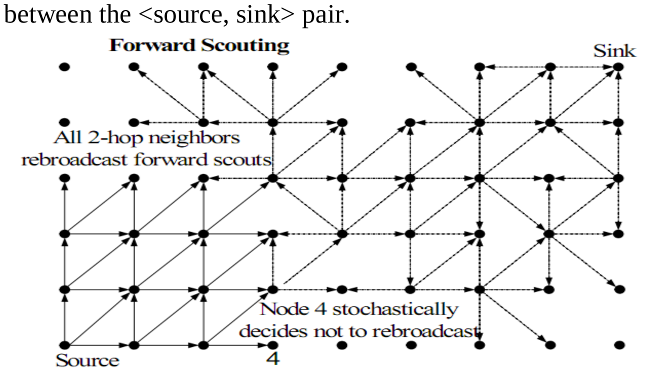


Fig.2  Forward scouting in BeeSensor with HL =3

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

A. Forward scouting

When an event1 is detected at a sensor node, it is handed over to a packer. The packer looks for an appropriate forager that might carry this event to a sink node. If the packer fails to find a forager, it launches a forward scout and encapsulates the event in its payload. A part from the agent's type field, the forward scout also carries four additional information fields in the header: scout ID, source node ID, minimum remaining energy (initialized to ∞)and the number of hops (initialized to zero). The forward scout is then broadcast to the neighbors of the source node. A forward scout does not know a priori the address of the sink node. A sink node interested in the event, carried in the payload of a scout, will convert the forward scoutto a backward scout. When an intermediate node i receives a forward scout from node j for the first time, it increments the hop field. The next step is to decide whether node i is going to broadcast the forward scout to the next hop or not. If node i is at j or less number of hops away from the source, it decides to rebroadcast the forward scout unconditionally. Otherwise, it rebroadcasts it with probability Pi [9]. It is proposed that HL should be a function of the current estimate of the hops between <source, sink> pair. In the next step, node compares the minimum remaining energy field to its own energy level and the value in the forward scout is updated to the minimum of the two.

B. Backward scouting

Nodes in BeeSensor maintain three types of tables: routing table, probability distribution table and forwarding table. Routing tables and probability distribution tables are maintained by source nodes only while forwarding tables are maintained by the sink and the intermediate nodes on a given path. When a sink node receives a forward scout, it extracts the event from the payload area and passes it to the application. Then it creates a new forwarding table entry which contains three fields: a unique path ID, next hop ID and previous hop ID .Next hop is set to the sink ID, previous hop entry in the forwarding table is set to the node ID from which the forward scout is received. The size of payload is truncated to zero and the minimum remaining energy field in the forward scout header is set to ∞. Then the sink node inserts a unique path ID to the header of the forward scout. Finally, it changes the agent ID to convert it to a backward scout. The backward scout is then forwarded to the node from which the forward scout was received. When a node j receives a backward scout from node j, it looks for a matching scout cache entry. If the information is found, it creates a forwarding table entry with the next hop set to i, path ID is set to the value contained in the header of the backward scout while previous hop is set to the previous hop ID present in the scout cache. The scout cache entry is then flushed. No future backward scout of the current generation is entertained and therefore the node will be a part of single route only. In other words, BeeSensor will discover node-disjoint paths only. Finally, it compares its own energy level with the level contained in the header of a backward scout and updates the field to the minimum of the two values. It then forwards the backward scout to the previous hop. Each intermediate node processes the backward scout in a similar way until it reaches the source node.

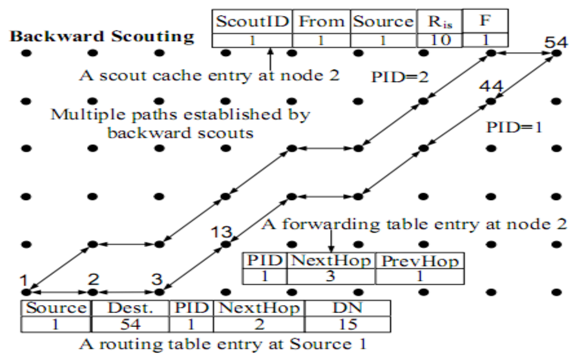


Fig.3 Backward scouting in SensorBee

Foraging

Like BeeAdHoc [8], foragers are the main workers in SensorBee as well. Their major role is to carry events to the sink nodes through a predetermined path that is selected stochastically at the source node. Foragers that follow the same path are grouped together in SensorBee. Foragers traverse using point-to-point mode by utilizing the forwarding information stored at intermediate nodes. They index the table using their path identifier (PID). Foragers also evaluate the quality of their path and report it back to fellow foragers at the source node.

Once a route is discovered, the foragers transport events to the sink node. The source node maintains a small event cache in which the events, generated during the route discovery process, are stored. A packer then selects a forager stochastically and encapsulates the event in it. Stochastic selection is based on probability distribution table.

Routing loops

The reward function in SensorBee is designed to provide loop freedom in the discovered routes Recall that backward scouts follow the maximum reward paths. Consequently, backward scouts keep moving in the direction of the source node reducing the probability of selecting a node which is at a larger distance than the current node Moreover, the forwarding table entry at a node indicates that the backward scout has already visited this node. Therefore, if a backward scout visits a node for the second time, it is dropped by the node and the corresponding entry is flushed. In this way, it is ensured that the discovered paths are loop free.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 2, Issue 8, August 2013*

## VII. PATH MAINTENANCE

Another important feature of BeeSensor, like Bee AdHoc protocol, is that it does not use explicit HELLO or route error (RERR) messages to check the validity of the routes. Swarming is simple but an elegant way of doing path maintenance. A path at a source node remains valid if it has foragers for it. The moment the dance number of a path in the routing table becomes zero, the path becomes invalid and therefore corresponding entry in the routing table is removed after FORG – the waiting time for the foragers that might be on their return journey towards the source node. However, if no forager arrives within the wait time, itis a clear indication that either the path is broken or the sink node is no more interested in the events. It is important to note here that forward scouting is only initiated if: (1) all the paths to a sink node are broken, and (2) events are still being generated / waiting in cache. Finally, it should also be noted that a TTL value is associated with foragers. If the forager is not used within this time, the forager dies. Forwarding table entries at intermediate nodes have also an associated lifetime.

## VIII. RESULTS:

### A. Total Energy Consumption

Wireless sensor nodes are equipped with a small non-rechargeable battery. In addition to this, due to immensely high number of nodes, it is practically infeasible to replace the battery of the nodes. Therefore, optimal utilization of available energy resource is critical to the overall operation of a sensor node and the network. Fig. 4 shows the total energy consumption of the protocols for both types of application scenarios. It is clear from Fig. 4 that Bee Sensor consumes least amount of total energy. Remember that nodes in BeeSensor located beyond 2hops perform stochastic broadcasting of scouts. As a result, number of transmissions in the network reduce and hence the energy consumption. Remember AODV is used as a benchmark algorithm for energy consumption because it is specially optimized for this purpose. Results show that in the static converge-cast scenario, energy consumed by AODV is either lower or equal than that of BeeSensor. This primarily is due to two reasons. First, the number of routediscoveries in converge-cast scenario is minimal as the sources do not change on run time. Second, average hop length between a source and the sink node is approximately 3.5 hops. Consequently, stochastic broadcasting impact is not much distinguishable. FP-Antas expected has the highest energy consumption and it remains approximately the same in both scenarios. It is important to note that FP-Ant is not a pure flooding protocol but it performs restrictive flooding of forward ants which carry the data ants as described One can easily extrapolate the energy consumption of a pure flooding algorithm.

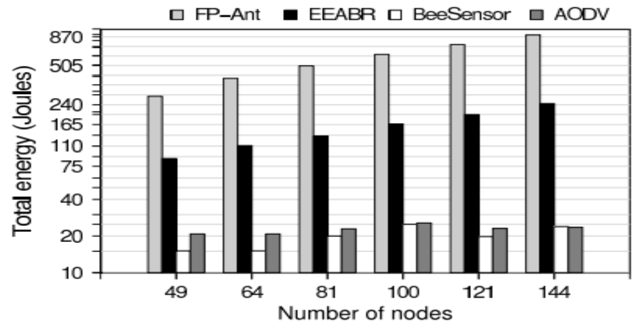


Fig.4 Total energy consumption comparison

### B. Energy Efficiency

The primary purpose of a routing protocol is to deliver a data packet from one node, called source node, in the network to another node, called a destination node. Rest of the design considerations are of secondary importance in various categories of the network. In case of wireless sensor networks, the secondary objective of a routing protocol is to route this data packet at minimal of energy cost. The energy efficiency metric is used to analyze the performance of the protocols under evaluation in meeting these two objectives. Energy efficiency values of these protocols are shown in Fig5. AODV and BeeSenso are the two best protocols in terms of energy efficiency. Best energy efficiency of BeeSensoris due to its low total energy consumption and high packet delivery ratio. Remember that if the loss ratio is high (or the packet delivery ratio is low), as in case of AODV, it shows that protocol is unable to establish a route quickly (i.e. in the first attempt).

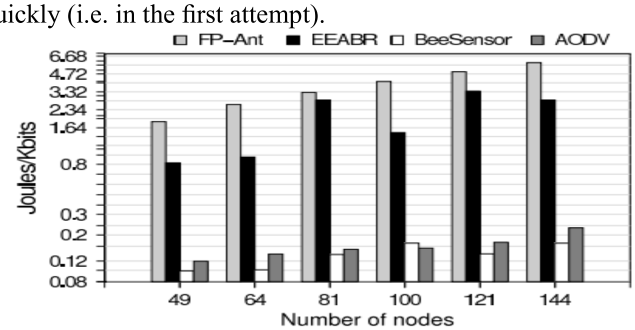


Fig.5 Energy efficiency comparison

## IX. CONCLUSION

In this paper , we have proposed a distributed, scalable and energy- efficiency bee-inspired routing protocol for MANETs– BeeSensor. Like other SI algorithms, BeeSensor is designed with the so called “bottom-up approach” in which the behavior of individual nodes is defined keeping in view of the desired network level behavior. However, in contrast to typical ACO-based algorithms, BeeSensor utilizes simple heuristic functions and allows complex stochastic routing function at the source nodes only. This not only results in fast switching of data packets but also provides relief to the low-end processors by reducing the processing overhead. In addition to this, BeeSensor discovers node-disjoint paths only which are not only fault-tolerant but also enable the protocol to consume the nodes battery at an equal rate.

The experimental results show that BeeSensor delivers superior performance in terms of packet delivery ratio and latency, but with the least energy consumption compared with other SI algorithms. The important reasons for this behavior of BeeSensor are: (1) a simple routing agent model, (2) agent-agent communication to discover optimal paths, (3) fixed size of route discovery agents that not only saves significant amount of energy during their transmission but also makes the algorithm scale to large networks, (4) distributed and decentralized control, and (5) self-organization to make it resilient to external failures.

## REFERENCES

[1] S. Mao and Y. T. Hou. Beamstar: *"An edge-based approach to routing in wireless sensor Networks*". Wireless Networks, 9:1284 – 1296, 2007

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. *"A performancecomparison of multi-hop wireless ad hoc network routing protocols"*. In Proceedings of
IEEE/ACM MobiCom, volume 1, Dallas, Texas, USA, October 1998.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *"A survey on sensorNetworks energy issue*". IEEE Communications Magazine, 40(8):102 – 114, August 2002.

[4]. H.F. Wedde, M. Farooq, T. Pannenbaecke, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat. *"BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks*
*inspired by bee behavior".* In Proceedings of Conference on Genetic and EvolutionaryComputation (GECCO), June, 2005.

[5]. Y. Zhang, L.D. Kuhn, and M.P.J. *Fromherz. Improvements on ant routing for sensornetworks*. In Proceedings of International Workshop on Ant Colony Optimization andSwarm Intelligence (ANTS), September, 2004.

[6]. G. Di Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An adaptive nature inspired algorithm for routing in mobile ad hoc networks. European Transactions onTelecommunications (ETT), Special Issue on Self Organization in Mobile Networking,

[7] C. E. Perkins and P. Bhagwat. *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*. In Proceedings of ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), New York, NY, 2004.

[8] A. A. Abbasi and M. Younis. *A survey on clustering algorithms for wireless sensor networks*. ACM Computer Communications, 30(14-15):2826 – 2841, October 2007.

[9]. W. Cai, X. Jin, Y. Zhang, K. Chen, and R. Wang. *ACO based QoS routing algorithm for wireless sensor networks*. In Proceedings of UIC, LNCS 4159, 2006.