A Biologically Inspired QoS Routing Algorithm for Mobile Ad Hoc Networks

Zhenyu Liu and Marta Z. Kwiatkowska
School of Computer Science
The University of Birmingham
Birmingham, England B15 2TT
{z.liu, mzk} @cs.bham.ac.uk

Costas Constantinou
Electronic Electrical and Computer Engineering
The University of Birmingham
Birmingham, England B15 2TT
c.constantinou@bham.ac.uk

Abstract

This paper presents EARA-QoS, a improved version of swarm-intelligence inspired ad hoc routing algorithm EARA introduced in [11]. In this algorithm, we use the principle of swarm intelligence to evolutionally maintain routing information. The biological concept of stigmergy is used to reduce the amount of control traffic. A light-weight QoS scheme is proposed to provide service-classified traffic control. The simulation results show that this novel routing algorithm performs well in a variety of network conditions.

1 Introduction

Mobile ad hoc networks (MANETs) are wireless mobile networks formed spontaneously. Communication in such a decentralised network typically involves temporary multi-hop relays, with the nodes using each other as the relay routers without any fixed infrastructure. This kind of network is very flexible and suitable for applications such as temporary information sharing in conferences, military actions and disaster rescues.

However, multi-hop routing, random movement of mobile nodes and other features unique to MANETs lead to enormous overhead for route discovery and maintenance. Furthermore, this problem is worsened by the resource constraints in energy, computational capacities and bandwidth.

Many approaches have been proposed to solve the routing challenge in MANETs, for example [10, 12, 13]. These previous works only provide a basic routing functionality that is sufficient for conventional applications such as file transfer or email download. To support applications such as VoIP in MANETs, which have a higher requirement for delay, jitter and packet losses, support for Quality-of-Service (QoS) is needed in addition to basic routing functionality. At present, the most fundamental challenges of QoS support in MANETs concern how to obtain the available bandwidth and maintain accurate values during dynamic evolution of such a network [8]. Inspired by common techniques for QoS provision in Internet, some researchers proposed the integration of QoS provision into the routing protocols [14,151. However, since these works implicitly assumed the same link concept as the one in wired networks, they still do not address the QoS problem for MANETs.

In this paper, we propose EARA-QoS, a new version of the self-organised Emergent Ad hoc Routing Algorithm (EARA) enhanced with QoS. EARA [11] is inspired by the foraging behaviour of biological ants. The biological concept of ***stigmergy*** is used to reduce the amount of control traffic. Local wireless medium information from the MAC layer is used as the artificial ***pheromone*** (a chemical used in ant communications) to reinforce optimal/sub-optimal paths without knowledge of the global topology. In addtion, this algorithm adopts the cross-layer design approach by using metrics from different layers to make routing decisions. These multi-criteria routing decisions allow for the better usage of network characteristics in selecting best routes among multiple available routes.

In EARA-QoS, in addition to the basic routing functionality, a light-weight QoS provision scheme is also integrated into the algorithm. In this scheme, traffic flows are classified into service classes based on their relative delay bounds. Delay sensitive traffic is given higher priority than other insensitive traffic flows. A token bucket queueing scheme is used to provide high priority to real-time traffic, and protect the lower-priority traffic from starvation. The results from simulation of mobile ad hoc networks show that this algorithm performs well over a variety of environmental conditions, such as network size, nodal mobility and traffic loads.

2 Background

In this section, we give a brief introduction to swarm intelligence and the parameters from the MAC layer.

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

2.1 Foraging Strategies in Ants

One famous example of biological swarm social behaviour is the ant colony foraging [3]. Many ant species have a trail-laying, trail-following behaviour when foraging: individual ants deposit a chemical substance called ***pheromone*** as they move from a food source to their nest, and foragers follow such pheromone trails. Subsequently, more ants are attracted by these pheromone trails and in turn reinforce them even more. As a result of this autocatalytic effect, the optimal solution emerges rapidly. In this food searching process a phenomenon called ***stigmergy*** plays a key role in developing and manipulating local information. It describes the indirect communication of individuals through modifying the environment.

From the self-organisation theory point of view, the behaviour of the social ant can be modelled based on four elements: positive feedback, negative feedback, randomness and multiple interactions [5]. This model of social ants using self-organisation theories provides powerful tools to transfer knowledge about the social insects to the design of intelligent decentralised problem-solving systems.

2.2 Quality-of-Service in MANETs

To support QoS in a network, the link characteristics such as bandwidth, delay, loss rate and error rate should be available and manageable. In wired networks, QoS provision relies on the possibility to make bandwidth reservations. Two distinct classes of QoS are implemented based on this assumption. Integrated Services (IntServ) [6] provides guaranteed bandwidth for flows, while Differentiated Services (DiffServ) **[4]** provides hard guarantees for service classes.

However, the situation in MANETs is completely different from that in wired networks. In wireless networks, the available bandwidth undergoes fast time-scale variations due to channel fading and errors from physical obstacles. These effects are not present in wired networks. Furthermore, the wireless channel is a shared-access medium, and the available bandwidth also varies with the number of hosts contending for the channel. Applying the concepts from wired QoS approaches directly to wireless ad hoc networks is difficult, because many assumptions in wired networks do not apply in wireless networks.

Consequently, IntServ based approaches are not applicable in MANETs. As an alternative to IntServ, the DiffServ approach classifies flows into several service classes whose packets are treated differently at the routing nodes. Although it is not practical to provide a hard separation of different service classes in MANETs, relative prioritisation is possible in such a way that traffic of a certain class is given a higher or lower priority than traffic of other service classes. One solution would be to divide the traffic into a predefined set of service classes that are defined by their relative delay bounds, such as delay sensitive ***(realtime)*** and insensitive ***(bulk)*** traffic. Realtime traffic should be given higher priority than bulk traffic. No absolute bandwidth grarantees are provided. Some work based on service differentiation rather than resource reservations already exists [2].

3 Description of EARA-QoS

EARA-QoS is an on-demand multipath routing algorithm for MANETs, inspired by the ant foraging intelligence. This algorithm incorporates positive feedback, negative feedback and randomness into the routing computation. Positive feedback originates from destination nodes to reinforce the existing pheromone on good paths. Ant-like packets, analogous to the ant foragers, are used to locally find new paths. Artificial pheromone is laid on the communication links between nodes and data packets are biased towards strong pheromone, but the next hop is chosen probabilistically. To prevent old routing solutions from remaining in the current network status, exponential pheromone decay is adopted as the negative feedback.

Each node using this algorithm maintains a ***probabilistic*** routing table. In this routing table, each route entry for the destination d is associated with a list of neighbour nodes j. A probability value $P_{i,j,d}$ in the list expresses the goodness of node j as the next hop to the destination d. For each neighbour, the shortest hop distance to the destination and the largest sequence number seen so far are also recorded.

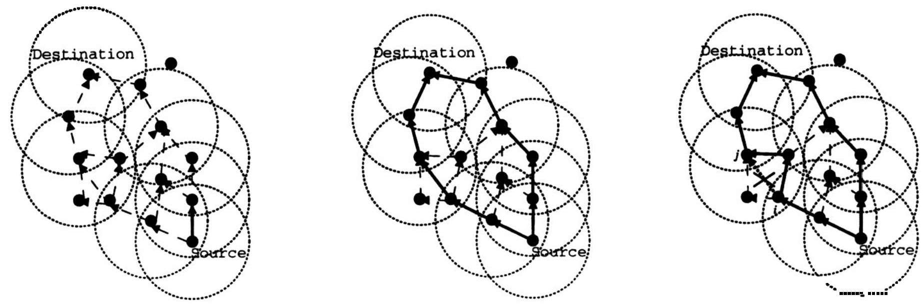
The routing table is updated as follows. In addition to the routing table, each node also possesses a ***pheromone*** table. This table tracks the amount of pheromone on each neighbour link. The table may be viewed as a matrix with rows corresponding to neighbourhood and columns to destinations. There are three threshhold values controlling the bounds on pheromone in the table. They are the ***upper pheromone*** $\mathcal{U}$ that prevents extreme differences in pheromone, the ***lower pheromone*** $\mathcal{L}$, below which data traffic cannot be forwarded, and the ***initial pheromone*** τ_0 that is assigned when a new route is found.

The routing probability value $P_{i,j,d}$ is computed by the composition of the pheromone values, the local heuristic values and the link delays as follows:

$$P_{i,j,d} = \frac{[\tau_{i,j,d}]^\alpha [\eta_{i,j}]^\beta [D_{i,j}]^\gamma}{\sum_{l \in \mathcal{N}_i} [\tau_{i,l,d}]^\alpha [\eta_{i,l}]^\beta [D_{i,l}]^\gamma} \qquad \tau_{i,j,d} > \mathcal{L} \quad (1)$$

where α, β and γ $(\alpha\beta, \gamma \geq 0)$ are three tunable parameters that control the relative weight of pheromone trail $\tau_{i,j,d}$, heuristic value $\eta_{i,j}$ and the link delay $D_{i,j}$. $\mathcal{N}_i$ is the neighbours of node i. With $\tau_{i,j,d} > \mathcal{L}$, data traffic can only be forwarded following a valid route.

The queue length at the MAC layer is a key factor that can affect the queueing delay in the buffer of node i, which is a measure of congestion in a node, or packet drop due to



(a) Initial Pheromone Setup (b) Path Reinforcement (c) Local Repair

Figure 1. Illustrating Working Mechanism of EARA-QoS

the size limit on the queue length. The heuristic value $\eta_{i,j}$ is calculated with the network interface queue length in node i:

$$\eta_{i,j} = 1 - \frac{q_{i,j}}{\sum_{l \in \mathcal{N}_i} q_{i,l}} \quad (2)$$

where $q_{i,j}$ is the length (in bits waiting to be sent) of the outgoing queue of the link $l_{i,j}$, $\mathcal{N}_i$ are the neighbours of node i.

Incorporating the heuristic value and link delay in the routing computation makes this algorithm possess the congestion awareness property. Based on the probabilistic routing table, data traffic will be distributed according to the probabilities for each neighbour in the routing table. The routing algorithm exhibits load balancing behaviour. Nodes with a large number of packets in the buffer are avoided.

The following control packets are used in EARA-QoS to perform routing computation:

- ***Route Request Packet*** (RQ) containing destination address, source address and broadcast ID.
- ***Route Reply Packet*** (RP) containing source address, destination address, sequence number, hop account and life-time.
- ***Reinforcement Signal*** (RS) containing destination address, pheromone value and sequence number.
- ***Local Foraging Ant*** (LFA) containing source address (the node that sent LFA), the least hop distance from the source to the destination, stack of intermediate node address and hop count.
- ***Hello Packet*** (HELLO) containing source (the node that sent Hello) address and hop count (set to 0).

The EARA-QoS algorithm consists of several components. Multiple loop-free routes are found with the ***route discovery*** procedure. EARA-QoS floods RQ packets to locate destination nodes. After the route discovery procedure sets up the initial pheromone trails to destination nodes, which is illustrated in Figure 1(a), destination nodes use RS packets as positive feedback signals to "pull" more data traffic through the good path(s) in the ***route maintenance*** procedure (refer to Figure 1(b)). This procedure is also useful to enable the local repair of degraded links. For instance, if an intermediate node i detects a link failure from one of its upstream links $l_{i,j}$, it can apply the reinforcement rules to discover an alternative path as shown in Figure 1(c).

In a dynamic network like MANET, the changes of the network topology create opportunities for new good paths to emerge and existing paths to fail. In order to make better use of this phenomenon, this algorithm launches LFA packets to locally search new routes whenever all the pheromone trails of a node towards some destination drop below the threshhold. The advantage of using LFA packets to locally search new routes is that this mechanism can save more control packets than the flooding technique and responds to the topology changes more quickly.

Route failures are handled with the ***local connectivity management.*** In case of a route failure occuring at node i, i sends a RS message that sets the ROUTE_RERR tag to inform upstream nodes of the link failure. This RS signal assigns to the corresponding links the lower bound $\mathcal{L}$. Here, RS plays the role of an explicit negative feedback signal to negatively reinforce the upstream nodes along the failure path. This negative feedback avoids causing buffer overflow due to caching on-flight packets from upstream nodes.

Moreover, the use of HELLO packets can also help to ensure that only nodes with bidirectional connectivity are deemed as neighbours. For this purpose, the HELLO packet sent by a node has an option to list the nodes from which it has heard HELLO packets, and nodes that receive the HELLO check to ensure that it uses only routes to neighbours that have sent HELLO packets.

3.1 QoS Considerations

This section describes a light-weight approach to DiffServ. The basic idea is to classify flows into a predefined set of service classes by their relative delay bounds. Admission control only works at the source node. There is no session or flow state information maintained at intermediate nodes.

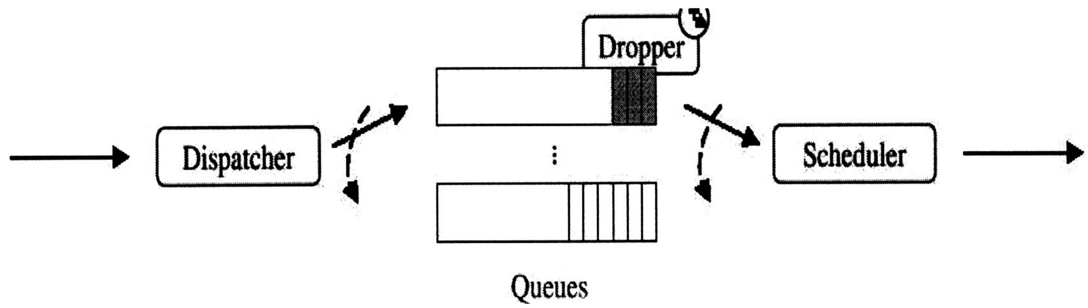


Figure 2. Overview of Service Differentiation Scheme

Once a realtime session is admitted, its packets are marked as RT (realtime service) and otherwise they are considered as best-effort bulk packets. As depicted in Figure 2, each of these traffic classes is buffered in a logically separate queue. A simple novel queueing strategy provides high priority to realtime traffic, and also protects the lower-priority traffic from starvation. No absolute bandwidth guarantees are provided in this scheme.

The queues are scheduled according to a ***token bucket scheme.*** In this scheme, prioritisation is achieved with token balancing. Each traffic class has a balance of tokens, and the class with higher balance has a higher priority when dequeuing the next packet for transmission. For each transmission of a packet of class s, an amount of w_s tokens is subtracted from the class's token balance and an equal fraction thereof is added to every other class's balance such that the sum of all tokens is always the same. The weight value w_s reflects the delay sensitivity assigned to the different classes. A higher weight value w_s corresponds to a lower delay sensitivity. The size of the token balance together with the value w_s determines the maximal length of a burst of traffic from one class. In this scheme, as long as the amount of delay-sensitive traffic does not grow too large, it is forwarded as quickly as possible, and if it does grow too large, starvation of other traffic classes is prevented. Setting the upper bound of a class's token balance depending on its delay-sensitivity enables further tuning of the described method.

In this packet schedule scheme, routing protocol packets are given unconditional priority before other packets. Moreover, realtime applications normally have stringent delay bounds for their traffic. This means that packets arriving too late are useless. From the applications point of view, there is no difference between late and lost packets. This implies that it is actually useless to forward realtime packets that stay in a router for more than a threshold amount of time, because they will be discarded at the destination anyway. Dropping those packets instead has the advantage of reducing the load in the network.

4 Preliminary Evaluation

To evaluate the performance of the EARA-QoS protocol, we carried out a series of simulations with the simulator *ns*-2 [1].

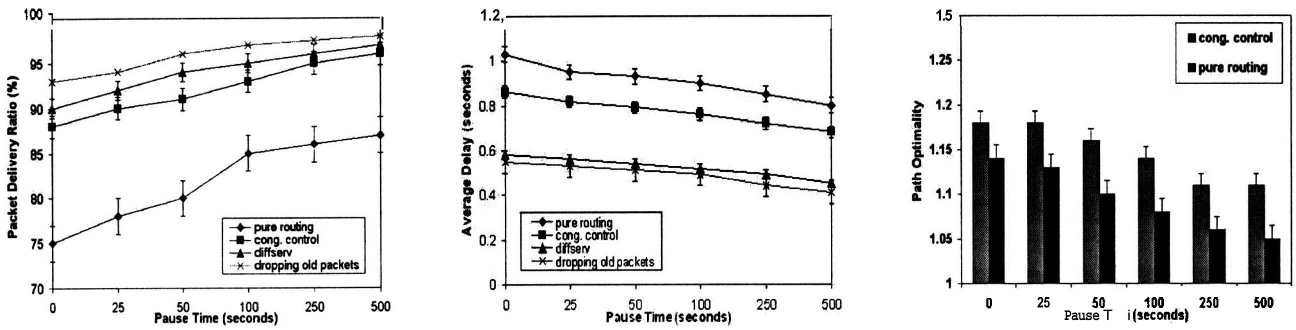
4.1 The Simulation Configurations

We use the IEEE 802.11 Distributed Coordination Function (DCF) as the MAC layer protocol. The radio model simulates Lucent's WaveLAN with a nominal bit rate of 2Mbps and a nominal transmission range of 250 meters. The radio propagation model is the two-ray ground model.

The mobility model we use is the ***Random Waypoint Model*** [7]. Each node independently chooses a random starting point and waits there for a duration called the ***pause time.*** It then randomly chooses a destination, and moves there at a velocity chosen uniformly between a minimum velocity v_{min} and a maximum velocity of v_{max}. When the node arrives at this destination, it again waits for the pause time and then moves to a new randomly chosen destination at a new randomly chosen velocity. Each node independently repeats this movement pattern through the simulation.

We performed two sets of simulations to study the performance of the routing algorithm in different load regimes. The first set was performed using 50 nodes in a rectangular field of 1000m x 1000m, and the other set was performed using 50 nodes in an area of 300m×3500m. In both sets, we define the v_{min} as 1 m/sec. For the first set, we define the v_{max} as 2 m/sec to simulate a conference scenario with participants wandering randomly. For the second set we define the v_{max} as 10m/sec to simulate vehicles travelling along a road. For both sets we use various pause times as the independent variable to simulate different mobility patterns as listed in the graphs.

The traffic consists of realtime and bulk packets. Realtime traffic is modelled as VoIP phone calls using bidirectional CBR sessions with a data rate ***9.6*** Kbit/s. The inter-arrival time of calls was exponentially distributed with a mean of 10 seconds. The length of a call was modelled



(a) Delivery Ratio vs Mobility (b) Delay vs Mobility (c) Path Opti. vs Mobility

Figure 3. Simulation of 50 nodes in an area of 1000m×1000m with v_{max} = 2 m/sec

according to a lognormal distribution with $\mu = 3.287$ and $\sigma = 0.891$, resulting in an average call length of about 40 seconds. According to [9], a delay of over 150 ms in a voice transmission is felt as disturbing by most users, and a delay above 250 ms is felt as unbearable. Thus in our simulations, real-time packets exceeding this maximum delay of 250 ms are considered to have arrived too late and are dropped. Bulk traffic was modelled as the transmission of a random amount of data with TCP NewReno, uniformly distributed between 100,000 and 5,000,000 bytes. The time between the initiation of data transfers was exponentially distributed with a mean of 30 seconds.

The different routing strategies simulated were:

- No congestion control and service differentiation.
- With congestion control (incorporating network interface queue length in routing computation).
- Congestion control and differentiation of realtime and bulk traffic, with $w_{realtime} = 1, w_{bulk} = 10$.
- Differentiation of realtime and bulk traffic, with $w_{realtime} = 1, w_{bulk} = 10$ and discarding of realtime packets queued for more than 0.1 seconds.

We evaluate the performance of our algorithm with four metrics, namely

- ***Packet delivery ratio,*** that is, the fraction of packets sent by a source node to that received by the corresponding destination node.
- ***Average*** ETE ***delay*** that reflects the total time needed to successfully deliver a packet by a source node till it is received by the corresponding destination node.
- ***Path optimality,*** that is, the fraction of delivered data packets that were routed by the protocol over routes of various lengths relative to the shortest optimal route (the shortest optimal route is determined by the simulator).

Each scenario uses the same simulation parameters as in [11]. In addition, we set γ as 1. The simulation time is 500 seconds and we executed 20 runs of the simulations.

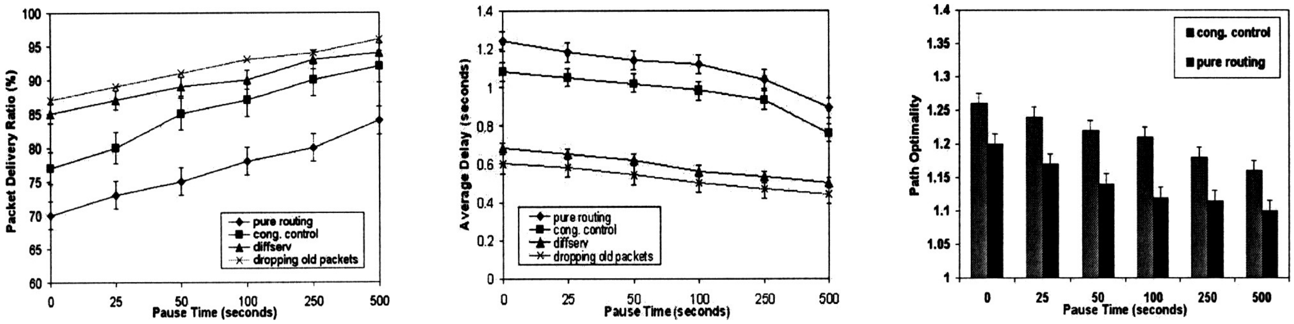
4.2 Results and Discussion

The results from the experiments are shown in Figure 3 and Figure 4. In these graphs, the error bars represent the standard deviation of the mean out of 20 runs.

From Figure 3(a), we see that the pure routing algorithm can deliver about 80% data at moderate nodal motion. This relatively good delivery is rooted in the multi-path nature of the algorithm. With congestion control, the packet delivery ratio can be improved by about 10%. This is due to the route discovery that avoids the congestion areas, which results many packets being delayed in queues and eventually dropped. Incorporating a differentiation scheme can also enhance the packet delivery ratio. This is mainly because the differentiation scheme prioritises service classes, which forwards as much realtime traffic as possible before the bulk traffic. Thus, realtime traffic traffic rarely drops due to blockage by bulk traffic. As the pause time decreases, which presents higher nodal motion, the data delivery ratio decreases for all the routig schemes. This indicates that our approach of service differentiation is limited by the mobility of nodes in a wireless ad hoc network. Route failures caused by the nodal motion account for a certain number of packet losses, which cannot be reduced by any of these approaches.

Figure 3(b) indicates that the packet delay is similar for the pure routing and congestion control routing, but integrating differentiation scheme with old realtime traffic droping can reduce much of the packet delay. This is because discarding packets that are delayed by more than 100 ms in an intermediate node's queue saves resources that benefit other packets, further improving the overall packet delivery ratio.

Figure 3(c) presents the path optimality. For the pure



(a) **Delivery Ratio** vs Mobility (b) **Delay** vs Mobility (c) **Path Opti.** vs Mobility

Figure 4. Simulation of 50 nodes in an area of 300m×3500m with v_{max} = 10 m/sec

routing scheme, the average path length is about 12% longer than the optimal one for the high nodal mobility and about 6% for the low nodal mobility. This indicates that, as nodal motion increases, the link errors increase. The local repair of the error links resulted in significant increase in suboptimal routes. For the congestion control scheme, the average route is about 5% longer than that of pure routing. This can be explained by the fact that the algorithm tries to build longer routes to avoid congestion areas.

In the second set of results shown in Figure 4, as both the network area and nodal mobility are greater than those in the first set, the average performance of the algorithm is slightly worse in terms of packet delivery ratio and transmission delay. When the nodal motion is low, the results of the two sets are close. As the nodal motion increases, the performance in the second set drops more than that of the first set. As the network area increases, the path optimality also increases slightly.

The limitation of this simulation presented here is that we choose all the algorithm parameters based on previous experience. Therefore, these preliminary simulation results cannot reflect the sensitivity of the algorithm. More elaborate experiments need to be done to investigate how to optimise the algorithm parameters and how these parameters are related to the network size and nodal mobility.

5 Conclusions

In this paper we present a biologically inspired routing algorithm for mobile multi-hop ad hoc networks. Through the concept of stigmergy, inspired by the biological ants, local optimal routes emerge without the global connectivity information. A light-weight DiffServ is integrated to provide simple QoS support. The results of early simulations show that this algorithm performs fairly well under situations of various nodal mobility, network density and data loads.

References

[1] A Collaboration between UC Berkeley, LBL, USC/ISI and Xerox PARC. *The ns Manual*, 2002. The VINT Project.

[2] G.-S. Ahn, A. T. Campbell, A. Veras, and L.-H. Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (swan). IEEE *Transactions on Mobile Computin* 1(3): 192–207, 2002.

[3] R. Beckers, J. L. Deneubourg, and S. Goss. Trails and u-turns in the selection of the shortest path by the ant lasius niger. *Journal of Theoretical Biology*, 159:397–415, 1992.

[4] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, 1998.

[5] E. Bonabeau, M. Dorigo, and G. Théraulaz. ***Swarm Intelligence: From Natural to Artificial Systems.*** Oxford University Press, 1999.

[6] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture. RFC 1633, 1994.

[7] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Yetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In ***Proceeding; of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)***, 1998.

[8] M. Gerharz, C. Waal, M. Frank, and P. James. A practical view on quality-of-service support in wireless ad hoc networks. In ***Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN),*** pages 185–196, 2003.

[9] ITU Recommendation G.114. ***One Way Transmission Time***, 1996.

[10] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. ***Mobile Computing***, 353, 1996.

[11] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A swarm intelligence routing algonthm for manets. In ***Proceedings of the IASTED International Conference on Communications Internet and Information Technology***, 2004.

[12] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. ***ACM SIGCOMM: Computer Communications Review***, 24(4):234–244, 1994.

[13] C. E. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In the ***Second IEEE Workshop on Mobile Computing systems and Applications***, pages 90–100, 1999.

[14] R. Ramanathan and M. Steenstrup. Hierachically-organized, multihop wireless networks for quality-of-service support. ***ACM Mobile Networks and Applications,*** 3(1):101–119, 1998.

[15] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: A core-extraction distributed ad hoc routing algorithm. IEEE ***Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks,*** 17(8):1454–65, 1999.

1550-445X/05 $20.00 © 2005 **IEEE**