A Biologically Inspired QoS Routing Algorithm for Mobile Ad Hoc Networks

Zhenyu Liu, Marta Z. Kwiatkowska, and Costas Constantinou

***Abstract*— This paper presents an Emergent Ad hoc Routing Algorithm with QoS provision (EARA-QoS). This ad hoc QoS routing algorithm is based on a swarm intelligence inspired routing infrastructure. In this algorithm, the principle of swarm intelligence is used to evolutionally maintain routing information. The biological concept of *stigmergy* is applied to reduce the amount of control traffic. This algorithm adopts the cross-layer optimisation concept to use parameters from different layers to determine routing. A lightweight QoS scheme is proposed to provide service-classified traffic control based on the data packet characteristics. The simulation results show that this novel routing algorithm performs well in a variety of network conditions.**

***Index Terms*— MANET, routing, QoS, swarm intelligence.**

I. 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 overheads for route discovery and maintenance. Furthermore, compared with the traditional networks, MANETs suffer from the resource constraints in energy, computational capacities and bandwidth.

To address the routing challenge in MANETs, many approaches have been proposed in the literature. Based on the routing mechanism for the traditional networks, the proactive approaches attempt to maintain routing information for each node in the network at all times [1]–[3], whereas the reactive approaches only find new routes when required [4]–[6]. Other approaches make use of geographical location information for routing [7], [8]. Those previous works only provide a basic “best effort” routing functionality that is sufficient for conventional applications such as file transfer or email download.

To support real-time applications such as VoIP and video stream in MANETs, which have a higher requirement for delay, jitter and packet losses, provision of Quality-of-Service (QoS) is necessary in addition to basic routing functionality. Given the nature of MANETs, it is difficult to support real-time applications with appropriate QoS. In some cases it may be even impossible to guarantee strict QoS requirements. But at the same time, QoS is of great importance in MANETs since it can improve performance and allow critical information to flow even under difficult conditions.

At present, the most fundamental challenges of QoS support in MANETs concern how to obtain the available bandwidth and maintain accurate values of link state information during the dynamic evolution of such a network [9]. Based on common techniques for QoS provision in the Internet, some researchers proposed the integration of QoS provision into the routing protocols [10], [11]. However, since these works implicitly assumed the same link concept as the one in wired networks, they still do not fully address the QoS problem for MANETs.

In this paper, we propose a new version of the self-organised Emergent Ad hoc Routing Algorithm with QoS provisioning (EARA-QoS). This QoS routing algorithm uses information from not only the network layer but also the MAC layer to compute routes and selects different paths to a destination depending on the packet characteristics. The underlying routing infrastructure, EARA originally proposed in [12], is a probabilistic multi-path algorithm inspired by the foraging behaviour of biological ants. The biological concept of *stigmergy* in an ant colony is used for the interaction of local nodes 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 the knowledge of the global topology.

One of the optimisations of EARA-QoS over EARA is the use of metrics from different layers to make routing decisions. This algorithm design concept is termed as the *cross-layer* design approach. Research [13] has shown the importance of cross-layer optimisations in MANETs, as the optimisation at a particular single layer might produce non-intuitive side-effects that will degrade the overall system performance. Moreover, the multiple-criteria routing decisions allow for the better usage of network characteristics in selecting best routes among multiple available routes to avoid forwarding additional data traffic through the congested areas, since the wireless medium over those *hotspots* is already very busy. The parameters for measuring wireless medium around a node depend largely on the MAC layer. In this paper, we focus on the IEEE 802.11 DCF mode [14], since it is the most widely used in both cellular wireless networks and in MANETs. This cross-layer technique of using MAC layer information can be applied

Z. Liu and M. Z. Kwiatkowska is with School of Computer Science, The University of Birmingham, Birmingham, England B15 2TT.

C. Constantinou is with the Department of Electronic Electrical and Computer Engineering, The University of Birmingham, Birmingham, England B15 2TT.

easily to other MAC protocols.

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

II. Background

In this section, we give a brief introduction to background knowledge on ant colony heuristics, and the QoS provision techniques in MANETs.

A. Foraging Strategies in Ants

One famous example of biological swarm social behaviour is the ant colony foraging [15] (see Figure 1). 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 auto-catalytic 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.

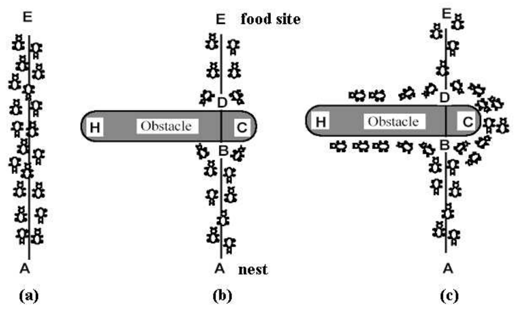


Fig. 1. All Ants Attempt to Take the Shortest Path

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 [16]. 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.

B. Quality-of-Service in MANETs

Quality-of-Service (QoS) provision techniques are used to provide some guarantee on network performance, such as average delay, jitter, etc. In wired networks, QoS provision can generally be achieved with the *over-provisioning of resources* and with *network traffic engineering* [17]. With the over-provisioning approach, resources are upgraded (e.g. fibre optic data link, advanced routers and network cards) to make networks more resistant to resource demanding applications. The advantage of this approach is that it is easy to be implemented. The main disadvantage of this approach is that all the applications still have the same priority, and the network may become unpredictable during times of bursting and peak traffic. In contrast, the idea of the traffic engineering approach is to classify applications into service classes and handle each class with a different priority. This approach overcomes the defect of the former since everyone is following a certain rule within the network.

The traffic engineering approach has two complementary means to achieve QoS provisioning, Integrated Services (IntServ) and Differentiated Services (DiffServ). IntServ [18] provides guaranteed bandwidth for flows, while DiffServ [19] provides hard guarantees for service classes. Both of the approaches rely on the possibility to make bandwidth reservations. The former was used in ATM (Asynchronous Transfer Mode) [20] and is today the method of achieving QoS in RSVP-IntServ [21]. On the other hand, in the DiffServ approach, no reservation is done within the network. Instead, QoS is achieved by mechanisms such as *Admission Control*, *Policy Manager*, *Traffic Classes* and *Queuing Schedulers*. These mechanisms are used to mark a packet to receive a particular forwarding or dropping treatment at each node.

Based on QoS provision techniques in wired networks, many QoS approaches are proposed to provide QoS services for MANETs. *Flexible QoS Model for MANETs* (FQMM) [22], is the first QoS approach for MANETs, which combines knowledge on IntServ/DiffServ in wired networks with consideration of MANETs. As an essential component to achieve the QoS provisioning, QoS routing algorithms tightly integrate QoS provisioning into routing protocols. The QoS version of AODV (QoS-AODV) [23], the *Core-Extraction Distributed Ad Hoc Routing* (CEDAR) protocol [10], the *Multimedia Support for Mobile Wireless Networks* (MMWN) protocol [11], and the ticket-based protocols [24] are examples of QoS routing algorithms proposed for MANETs.

On the other hand, QoS signaling techniques are independent of the underlying routing protocols. The *In-band Signalling for QoS in Ad-Hoc Mobile Networks* (INSIGNIA) algorithm [25] is the typical signaling protocol designed exclusively for MANETS.

The idea of CEDAR, MMWN, and ticket-based protocols is to disseminate link-state information across the network in order to enable other nodes to find routes that meet certain QoS criteria, like the minimum bandwidth. On the other hand, INSIGNIA piggybacks resource reservations onto data packets, which can be modified by intermediate nodes to inform the communication endpoint nodes in case of lack of resources. All those approaches are based on the idea that the wireless links between mobile nodes have certain QoS related properties, in particular a known amount of available bandwidth, and that nodes are able to give guarantees for traffic traversing these links.

III. Critique of Existing QoS Approaches in MANETs

Nowadays, most of the QoS provisioning techniques are derived from the QoS approaches of the wired networks. However, QoS support approaches proposed in wired networks are based on the assumption that the link characteristics such as bandwidth, delay, loss rate and error rate must be available and manageable. However, given the challenges of MANETs, e.g. dynamic topology and time-varying link capacity, this assumption does not apply any longer. Thus, applying the concepts of wired traffic engineering QoS approaches directly to MANETs is extremely difficult.

Generally, the situation in MANETs is completely different from those 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. In MANETs, the wireless channel is a shared-access medium, and the available bandwidth even varies with the number of hosts contending for the channel. Below we analyse why the IntServ/DiffServ models are not appropriate for MANETs respectively.

IntServ based approaches are not applicable for MANETs mainly due to two factors, *huge resource consumption* and *computation power limitation*. Firstly, to support IntServ, a huge amount of link state information has to be built and maintained for each mobile node. The amount of state information increases proportionally with the number of flows, which is also a problem with the current IntServ QoS scheme. Secondly, current wireless networks employ two major MAC techniques, the single-channel approach and the multiple channel approach. With single-channel approach (e.g. IEEE 802.11 [14]), all nodes share the same channel and therefore potentially interfere with each other. With a multiple-channel approach (e.g. Bluetooth [26] or CDMA [27]), nodes can communicate on several channels simultaneously. Both of the two MAC techniques have a similar bandwidth reservation mechanism. This common mechanism requires a transmission schedule to define time slots, in which nodes take their turns periodically. For each slot, its duration and a set of possible simultaneous transmissions must be defined. However, in wireless networks, the problem of finding an optimal schedule is proved to be NP-complete [28], which is a fundamental limitation of QoS provisioning in wireless networks.

On the other hand, the DiffServ approach is a lightweight QoS model for interior routers since individual state flows are aggregated into sets of service classes whose packets are treated differently at the routing nodes. This makes routing a lot easier in the network. Thus this approach could be a potential solution for MANETs. Even though 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 guarantees are provided. Some work based on service differentiation rather than resource reservations in MANETs already exists [29].

IV. 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.

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 threshold 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.

In addition to the routing data structures present above, 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 lifetime.
- *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).

A. Parameters of Lower Layers

1) The Average MAC Layer Utilisation: The first metric is the average MAC layer utilisation ϕ_i for a node i. This metric measures the usage of the wireless medium around that node. As the instantaneous MAC layer utilisation at a node is either 1 (busy) or 0 (idle), we average this value over a period of time window T_{mac} as follows:

$$\phi_i = \frac{\sum T_{busy}}{T_{mac}}, \qquad T_{busy} \subseteq T_{mac} \tag{1}$$

where T_{busy} is the time when the medium is busy in the window T_{mac}. This average MAC utilisation ϕ_i indicates the degree to which the wireless medium around that node is busy or idle.

We consider the instantaneous MAC layer utilisation level at a node to be 1 when the wireless medium around that node either detects physical carrier to be present or is deferring due to virtual carrier sensing, inter-frame spacing, or backoff. In addition, we also consider the medium is busy at any time when the node has at least one packet in the transmission queue.

2) The Transmission Queue Heuristic: The second metric is a heuristic value η_i that is calculated with the network interface transmission queue length in the current node i. Apart from the media status, the transmission queue length is also a key factor that can affect the packet latency or packet drop due to the size limit on the queue length. We define the heuristic value with the following rules. If the outgoing network interface employs a single queue scheme, the heuristic value η_i is defined as:

$$\eta_i = \frac{q_i}{q_{max}} \tag{2}$$

where q_i is the length (in bytes waiting to be sent) of the interface queue in node i, and q_{max} is the maximum packet bytes allowed in the queue.

If the network interface employs the multiple virtual queue scheme for each outgoing link, the heuristic value $\eta_{i,n}$ is defined as:

$$\eta_{i,n} = 1 - \frac{q_{i,n}}{\sum_{l \in \mathcal{N}_i} q_{i,l}} \tag{3}$$

where $q_{i,n}$ is the length (in bytes waiting to be sent) of the virtual queue of the link $l_{i,n}$ in node i and $\mathcal{N}_i$ denotes the neighbourhood of node i as a next-hop to some destination.

3) The Average MAC Layer Delay: The last metric is the MAC layer delay $D_{i,j}$ for the link $l_{i,j}$. The MAC layer delay $D_{i,j}$ is defined as the interval from when the RTS frame is sent at node i to when the data frame is received successfully at node j. The average MAC delay $\tilde{D}_{i,j}$ is obtained by averaging these values over a time window T_{cont} as follows:

$$\tilde{D}_{i,j} = \frac{a}{\sum D^t_{i,j}}, \qquad t \subseteq T_{mac} \tag{4}$$

where t is the time interval in the window T_{cont}, and a is a coefficient.

This average MAC delay $\tilde{D}_{i,j}$ indicates the degree of interference. In regions where there is a lot of interference from other nodes, MAC delay is high due to the contention of the channel.

B. Data Propagation

When multiple virtual queue scheme is employed, 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:

$$P_{i,j,d} = \frac{[\tau_{i,j,d}]^\alpha [\tilde{D}_{i,j}]^\beta [\eta_{i,j}]^\gamma}{\sum_{l \in \mathcal{N}_i} [\tau_{i,l,d}]^\alpha [\tilde{D}_{i,j}]^\beta [\eta_{i,l}]^\gamma}, \qquad \tau_{i,j,d} > \mathcal{L} \tag{5}$$

where α β, and γ ($\alpha, \beta, \gamma \geq 0$) are tunable parameters that control the relative weight of pheromone trail $\tau_{i,j,d}$, MAC delay $\tilde{D}_{i,j}$ and heuristic value $\eta_{i,j}$, and $\mathcal{N}_i$ is the neighbourhood as a next-hop to some destination d.

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 EARA-QoS algorithm consists of several components. They are the *route discovery* procedure, the positive and negative *reinforcement*, and the *local connectivity management*.

C. Route Discovery

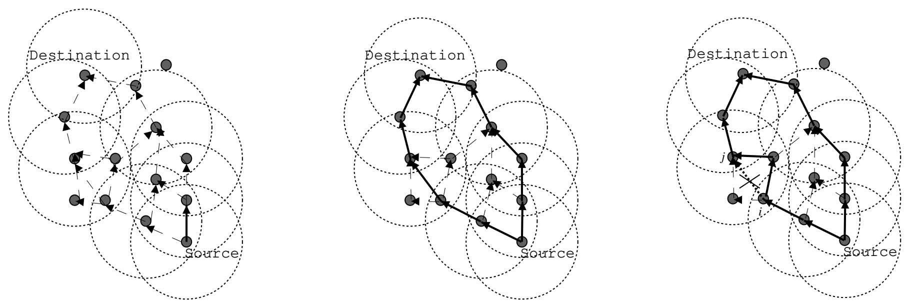
We use a similar route discovery procedure as described in [12]. On initialisation, a neighbourhood for each node is built using the single-hop HELLO messages. Whenever a traffic source s needs a route to a destination d, it broadcasts *route request* packets (RQ) across the network. Rather than simply flooding the RQ packets, we adopt the probabilistic broadcast scheme explored in [30] combined with the MAC layer utilisation ϕ. When a node first receives a packet, with probability p' it broadcasts the packet to its neighbours, and with probability $1 - p'$ it discards the packet. The probability value p' is calculated as

$$p' = e^{-\sigma\phi} \tag{6}$$

where σ ($\sigma > 0$) is the coefficient. This broadcast scheme helps to discover new routes avoiding congestion areas, but at the cost of missing potential routes to the destination.

During the course of flooding RQ packets to the destination d, the intermediate node j receiving a RQ packet first sets up reverse paths to the source by recording the source address and the previous hop node in the *message cache*. If a valid route to the destination d is available, that is, there is at least one link associated with the pheromone trail greater than the lower pheromone bound $\mathcal{L}$, the intermediate node j generates a *route reply* (RP). The RP is routed back to the source s via the reverse paths. Otherwise, the RQ is rebroadcast.

Other than just establishing a single forward path, when the destination node d receives RQs it will send a RP to *all* the neighbours from which it sees a RQ. In order to maintain multiple loop-free paths at each intermediate node j, node



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

Fig. 2. Illustrating Working Mechanism of EARA-QoS

j must record all new forward paths that possess the latest sequence number but hold a lower hop-count in its routing table, and also send a RP to all the neighbours from which it saw a RQ. During the course of the RP tracking back to the source s, an initial pheromone value τ_0 is assigned to the corresponding neighbour node, which indicates a valid route to the destination d. This process is illustrated in Figure 2(a).

D. Route Reinforcement

After the destination node d receives the data traffic sent by the source node s, it begins to *reinforce* some good neighbour(s) n in order to "pull" more data traffic through the good path(s) by sending *reinforcement signal* packets (RS) whenever it detects new good paths. When node n receives a RS, it knows it has an outgoing link toward the destination d, which is currently deemed a good path. Subsequently, node n updates the corresponding pheromone table entry with the value $\Delta\tau$ and forwards a RS packet to (at least one) selected neighbour locally based on its message cache, e.g. the neighbour(s) that saw the least hops of the incoming packets.

The amount of the pheromone $\Delta\tau$ used to positively reinforce the previous hop neighbour is computed as follows. If the RS packet π is sent by the destination to node n, then $\Delta\tau^{\pi}_{n,d}$ is calculated using the upper bound pheromone value $\mathcal{U}$,

$$\Delta\tau^{\pi}_{n,d} = \mathcal{U} \cdot e^{-(\alpha'\Delta h_s + \beta'\hat{n})} \tag{7}$$

If the RS packet π is sent by an intermediate node j towards node i, the $\Delta\tau^{\pi}_{i,j,d}$ is calculated using the current largest pheromone value $\max(\tau_{j,k,d})$ in node j with the next hop k to the destination d in the pheromone table,

$$\Delta\tau^{\pi}_{i,j,d} = \max(\tau_{j,k,d}) \cdot e^{-(\alpha'\Delta h_{s,j} + \beta'\hat{n}_{j,d} + \gamma'\eta_j)} \tag{8}$$

where α', β' and γ' are parameters that control the relative weight of the relative source hop distance $\Delta h_{s,j}$, the relative packet number $\hat{n}_{j,d}$ and the local queue heuristic η_j. Incorporating the congestion-measuring metric η_j into the reinforcement can lead data traffic to avoid the congestion areas.

The relative source hop distance $\Delta h_{s,j}$ is calculated as follows:

$$\Delta h_{s,j} = h^i_{s,j} - h^{min}_{s,j} \tag{9}$$

where $h^i_{s,j}$ is the shortest hop distance from the source s to the current node j through node i, and h^{min}_s is the shortest hop distance from s to j. This parameter is used to ensure that paths with shorter hop distance from the source node to the current node are reinforced with more pheromone.

The relative packet number $\hat{n}_{j,d}$ is calculated as follows:

$$\hat{n}_{j,d} = 1 - n^i_{j,d}/n_{j,d} \tag{10}$$

where $n^i_{j,d}$ is the number of incoming packets from neighbour i to the destination d, and $n_{j,d}$ is the total number of incoming packets towards the destination d. This parameter is used to indicate that the data forwarding capacity of a link also affects the reinforcement. The more data arrives, the stronger reinforcement is generated for the corresponding link.

On receiving the RS from a neighbour n, node i needs to positively increase the pheromone of the link towards node d. If the sequence number in the RS is greater than the one recorded in the pheromone table, node i updates its corresponding pheromone $\tau_{i,n,d}$ with the value of $\Delta\tau^{\pi}_{i,n,d}$ carried on the RS π:

$$\tau_{i,n,d} := \Delta\tau^{\pi}_{i,n,d} \tag{11}$$

If the sequence number is equal to the current one, then:

$$\tau_{i,n,d} := \begin{cases} \Delta\tau^{\pi}_{i,n,d}, & \text{if } \tau_{i,n,d} < \Delta\tau^{\pi}_{i,n,d} \\ \tau_{i,n,d}, & \text{otherwise} \end{cases} \tag{12}$$

If the sequence number in RS is less than the current one in the pheromone table, then this RS is just discarded.

Node i also has to decide to reinforce (at least) one of its neighbours by sending the RS message based on its own message cache. This process will continue until reaching the source node s. As a result of this reinforcement, good quality routes emerge, which is illustrated in Figure 2(b). The same procedure can apply to any intermediate node to perform local link error repair as long as it has pheromone value that is greater than the lower bound $\mathcal{L}$. 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 2(c).

There is also an implicit negative reinforcement for the pheromone values. Within every time interval T_{decay}, if there is no data towards a neighbour node, its corresponding pheromone value decays by a factor ρ as follows:

$$\tau_{i,n,d} := (1-\rho) \cdot \tau_{i,n,d}, \qquad \rho \in (0.1] \tag{13}$$

E. Local Foraging Ants

In a dynamic network like MANET, the changes of the network topology create chances for new good paths to emerge. In order to make use of this phenomenon, this algorithm launches *local foraging ants* (LFA) with a time interval T_{ant} to locally search for new routes whenever all the pheromone trails of a node towards some destination drop below the threshold τ_0. The LFA will take a random walk from its original node.

During the course of its walk, if the LFA detects congestion around a node (the average channel utilisation ϕ is greater than a predefined threshold value ϕ'), then the LFA dies to avoid increasingly use the wireless medium. Otherwise, the LFA pushes the address of the nodes that it has travelled into its memory stack $\mathcal{S}$. To avoid forming of loops, LFA will not choose to travel to the node that is already in $\mathcal{S}$. Before reaching the maximum hop, if LFA can find a node with pheromone trails greater than τ_{ant} and the hop distance to destination not greater than the one from its original nest, it returns to its 'nest' following its memory stack $\mathcal{S}$ and updates the corresponding paths with τ_0. Otherwise, it simply dies.

F. Local Connectivity Management

Nodes maintain their local connectivity in two ways. Whenever a node receives a packet from a neighbour, it updates its local connectivity information to ensure that it includes this neighbour. In the event that a node has not sent any packets to its neighbours within a time interval T_{hel}, it has to broadcast a HELLO packet to its neighbours. Failure to receive packets from the neighbourhood in T_{hel} indicates changes in the local connectivity. If HELLO packets are not received from the next hop along an active path, the node that uses that next hop is sent notification of link failure.

In case of a route failure occurring at node i, i cannot forward a data packet to the next hop for the intended destination d. Node i sends a RS message that sets 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.

G. The QoS Provision Scheme

This section describes a lightweight 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. 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 3, each of these traffic classes is buffered in a logically separate queue. A simple novel queuing strategy, based on the token bucket scheme, provides high priority to realtime traffic, and also protects the lower-priority traffic from starvation. No absolute bandwidth guarantees are provided in this scheme. We explain this queuing strategy and its novelty below.

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' token balance and an equal fraction thereof is added to every other class' 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' token balance depending on its delay-sensitivity enables further tuning of the described method.

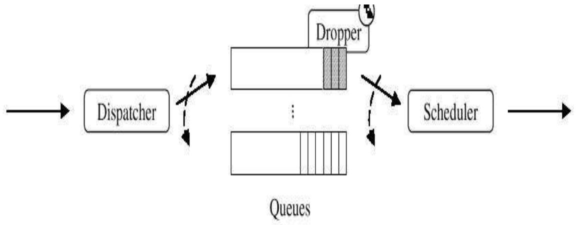


Fig. 3. Overview of Service Differentiation Scheme

In this packet scheduling 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 application's 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. To our knowledge, this service classification based queuing scheme is the simplest implemented QoS provisioning technique designed exclusively for MANETs so far.

V. Characteristics of the Algorithm

This proposed protocol, implementing the cross-layer design concept, exhibits some properties that show its fitness as a solution for mobile ad hoc networks:

- Loop-freeness: during the route discovery phase, the nodes record the unique sequence number of RP packets.

If a node receives a duplicate packet, it drops the packet and does not update the pheromone table, so they do not generate a loop. In addition, the ant exploration procedure also prevents loops by using the memory stack.

- Local work: no routing tables or other information blocks have to be transmitted to neighbours or to all the nodes of the network, and decision rules operate on the basis of locally available information.
- Support for multiple paths: each node has a routing table with entries for all its neighbours recording the relative pheromone concentration. This method provides possible support for multiple-path routing decisions.
- Congestion awareness: since this algorithm adopts the cross-layer optimisation concept to incorporate parameters, measuring wireless medium conditions from lower layers in routing computation, the selection of routes through congestion areas are avoided.
- Inherent routing oscillation avoidance: the probabilistic multi-path data forwarding scheme of EARA-QoS *disperses* the data traffic across a set of valid paths. This feature helps to avoid the oscillation between congested and uncongested states in routing selection, which occurs frequently for the single path QoS routing algorithms.
- Lightweight QoS provisioning: the transmission delay related service classification and the token bucket queue scheduling scheme provide an easily implementable lightweight QoS support for real-time applications.

VI. Preliminary Evaluation

To evaluate the performance of the EARA-QoS protocol, we carried out a series of simulations with the simulator *ns*-2 (version 2.27) [31].

A. 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* [32]. 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×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 10 m/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 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 [33], 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 in buffers 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.
- *Average delay jitter*, that is, the amount of variation in the end-to-end transit delays. This metric is crucial in terms of QoS to real-time applications such as video and audio streaming.
- *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).

The simulation time is 500 seconds. Each scenario uses the same protocol parameters as illustrated in Table I. A point to be noted here is that all the protocol parameters in Table I are tunable to fit different situations. For instance, when the parameter T_{ant} is increased, the response to the link changes will consequently improve, which results in swifter congestion and link error detection, and consequently faster data delivery. However, this improvement is largely at the cost of the bandwidth, since both the routing overheads and data delivery require large bandwidth. Therefore, the adjustment of protocol parameters trades off between network resources and algorithm performance. The pause time, which affects the

TABLE I
EARA-QoS PARAMETER SETTINGS

Parameters	Value
Upper Pheromone Bound $\mathcal{U}$	50.0
Lower Pheromone Bound $\mathcal{L}$	0.1
Initial Pheromone Value τ_0	1.0
Pheromone Trail Weight α	1.0
MAC Delay Weight β	1.0
Shortest Hop Weight α'	1.0
Relative Packet Weight β'	1.0
Reinforcement Queue Weight γ'	1.0
Pheromone Decay Factor ρ	0.05
MAC Delay Coefficient a	1.0
Flooding Coefficient σ	1.0
Route Error Timer T_{err}	5.0 sec
Ant Sending Interval T_{ant}	1.0 sec
Contention Window T_{mac}	10.0 sec
Pheromone Decay T_{decay}	1.0 sec
Hello Timer T_{hello}	1.0 sec
Route Discovery Timeout	1.0 sec
Route Expiration Timeout	10.0 sec
Reverse Route Timeout	3.0 sec
Number of Retransmissions	10

motion of nodes defined in the random waypoint mobility model, is varied as 0, 25, 50, 100, 250, 500 seconds. With pause time at 0 seconds, all nodes in the network move continually, while with pause time at 500 seconds, all nodes are nearly stable throughout the simulation.

B. Results and Discussion

The results from the experiments are all of confidence interval of 95%. In the graphs, the error bars represent the standard deviation of the mean out of 20 runs.

We first discuss the robustness of the algorithm in terms of packet delivery ratio. The first set of simulation is performed in an area of $1000m \times 1000m$ with 50 nodes, with results in Figure 4. We see that the pure routing algorithm can deliver about 80% of 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 in 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 real-time 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 routing 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.

Next, we scale the scenario into a more difficult one, with 100 nodes in a rectangular area of $300m \times 3500m$. The results of the simulation in Figure 5 show the same trend: as the nodal mobility gets more intensive, the performance of the algorithms gets worse. When the nodal motion is low, the

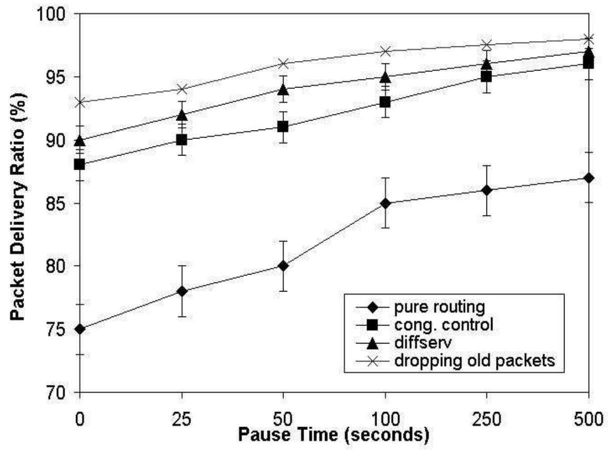


Fig. 4. Simulation of 50 nodes in an area of 1000m×1000m with $v_{max} = 2$ m/sec

results of the two sets are close. As the nodal motion increases, the performance in the second set drops more drastically than that of the first set.

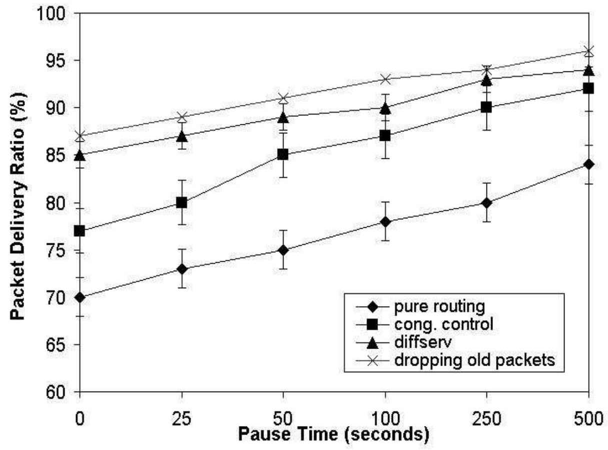


Fig. 5. Simulation of 100 nodes in an area of 300m×3500m with $v_{max} = 10$ m/sec

Secondly, we investigate the quality of service provided by the protocols with respect of average delay. The results of the first set of simulations in Figure 6 indicate that the packet delay is similar for the pure routing and congestion control routing, but integrating differentiation scheme with old realtime traffic dropping can reduce much of the packet delay. This is because discarding packets that are delayed by more than 250 ms in an intermediate node's queue saves resources that benefit other packets, further improving the overall packet delivery ratio. However, for the case of continual nodal movement (the pause time is 0 seconds), the average delay for the class differentiation routing is around 400 ms, that is, the maximum delay allowed for most of the real-time applications, which leads to more packets being dropped. When the scenario becomes difficult as the second set of simulations shows, the performance of the continual movement becomes a bit worse. This indicates that real-time applications suffer significant

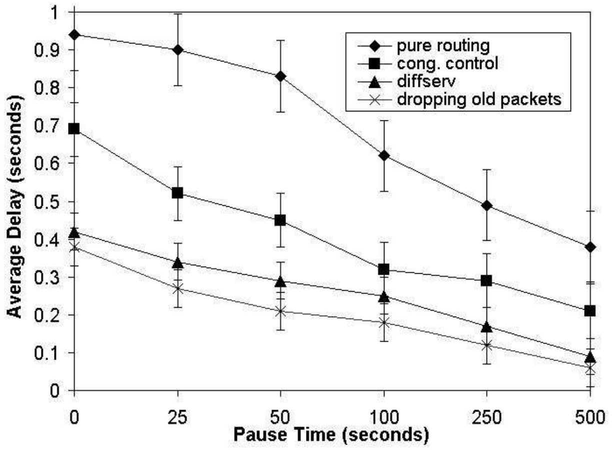


Fig. 6. Simulation of 50 nodes in an area of 1000m×1000m with $v_{max} = 2$ m/sec

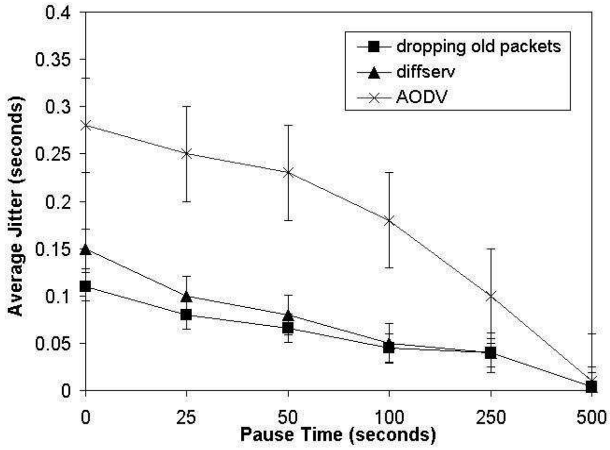


Fig. 8. Simulation of 50 nodes in an area of 1000m×1000m with $v_{max} = 2$ m/sec

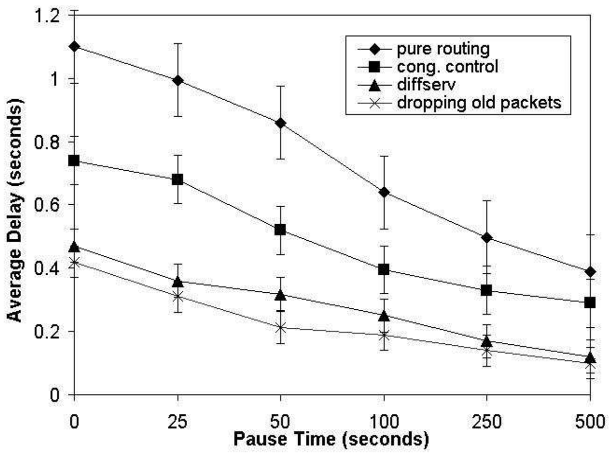


Fig. 7. Simulation of 100 nodes in an area of 300m×3500m with $v_{max} = 10$ m/sec

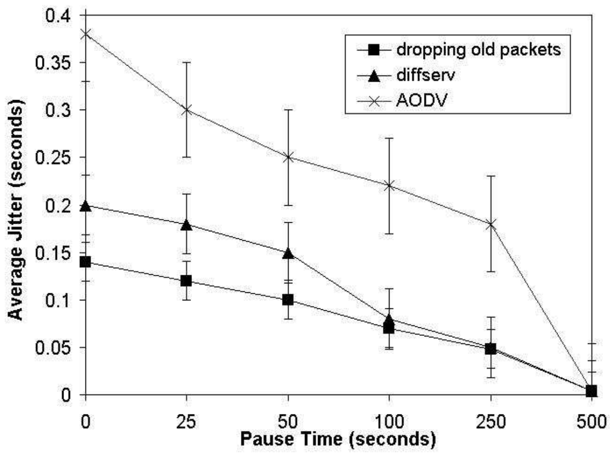


Fig. 9. Simulation of 100 nodes in an area of 300m×3500m with $v_{max} = 10$ m/sec

transmission delays under the intense movement situations.

Figure 7 shows the results of the second set of simulations. As the conditions become more difficult, the performance of the pure routing and congestion control routing drops more drastically. However, the performance of integrating differentiation scheme routing can still remain close to that of the first set due to the priority classification and over-delayed packet dropping.

Thirdly, we discuss the metric of average delay jitter. We compare the service differentiation routing against the AODV protocol. The results of the simulation are shown in Figure 8 and 9. The bad jitter of AODV is mainly because AODV relies on just one path. When this path becomes inefficient or invalid, a small number of packets with very high delay can lead to high delay variability. On the other hand, EARA-QoS is featured with multi-path routing. Packets are sent over different paths, rather than a single optimal path, which helps to reduce the jitter when a single path becomes bad. Moreover, the dropping of over-delayed packets in service differentiation routing further improves the performance in terms of jitter. When the situation gets even more difficult, the performance of the service differentiation service still remains stable.

Finally, we discuss the metric of path optimality. Figure 10 presents the path optimality. For the pure 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 the nodal motion increases, the link errors increase. The local repair of the error links resulted in significant increase in sub-optimal 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. Figure 11 shows the results of the second set of simulations. As the situation gets hard, the path optimality also increases slightly. This is because the more difficult the situation, the more link errors happen, which leads to longer alternative paths evolving.

One limitation of this simulation experiment is that we choose all the algorithm parameters based on previous experience. Therefore, these preliminary simulation results cannot

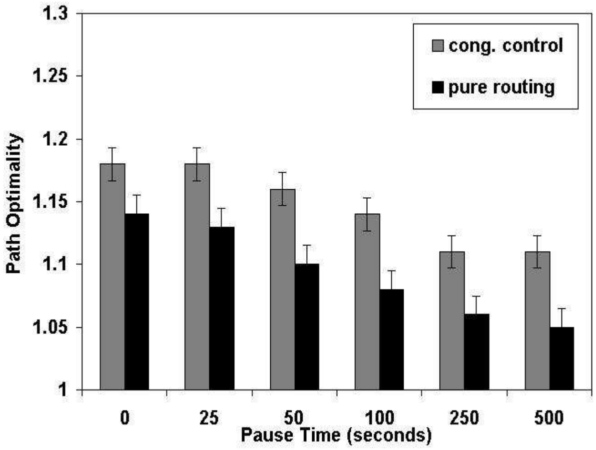


Fig. 10. Simulation of 50 nodes in an area of 1000m×1000m with $v_{max} = 2$ m/sec

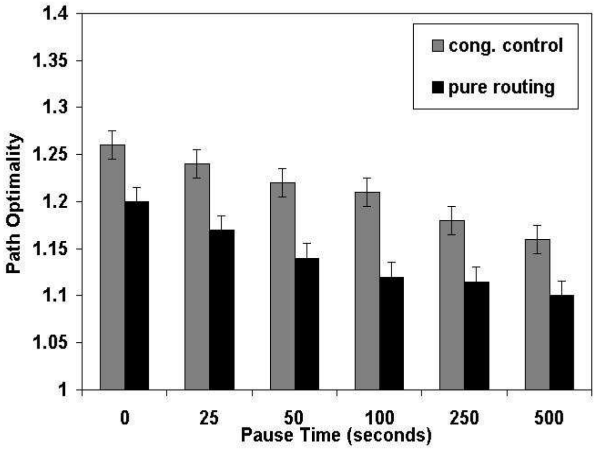


Fig. 11. Simulation of 100 nodes in an area of 300m×3500m with $v_{max} =$ 10 m/sec

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 metrics such as the network size, nodal mobility and data loads.

VII. 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. By adopting the cross-layer optimisation concept, both the network layer and the MAC layer information are used to compute routes that avoid the congested areas. Moreover, a lightweight DiffServ is integrated to provide QoS provisioning exclusively for MANETs. The core of this QoS provisioning technique is the service class differentiation based queuing scheme. The results of simulation experiments show that this algorithm performs fairly well under situations of various nodal mobility, network density and data loads.

References

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM: Computer Communications Review*, vol. 24, no. 4, pp. 234–244, 1994.

[2] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimised link state routing protocol," IETF MANET, Internet Draft, 1998.

[3] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proceedings of the 1st annual international conference on Mobile computing and networking*. ACM Press, 1995, pp. 86–95.

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

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

[6] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 1997, pp. 1405–1413.

[7] Y. B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," in *Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998, pp. 66–75.

[8] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proceedings of the IEEE/ACM international Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998, pp. 76–84.

[9] 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)*, 2003, pp. 185–196.

[10] 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*, vol. 17, no. 8, pp. 1454–65, 1999.

[11] R. Ramanathan and M. Steenstrup, "Hierachically-organized, multihop wireless networks for quality-of-service support," *ACM Mobile Networks and Applications*, vol. 3, no. 1, pp. 101–119, 1998.

[12] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou, "A swarm intelligence routing algorithm for manets," in *Proceedings of the IASTED International Conference on Communications Internet and Information Technology*, no. 433, November 2004, pp. 484–489.

[13] E. M. Royer, S. J. Lee, and C. E. Perkins, "The effects of mac protocols on ad hoc networks communication," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2, 2000, pp. 543–548.

[14] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Institute of Electrical and Electronics Engineers Std., 1997.

[15] 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*, vol. 159, pp. 397–415, 1992.

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

[17] S. Vegesna and S. Vegesna, *IP Quality of Service*, 1st ed. Cisco Press, 2001.

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

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

[20] M. D. Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, 3rd ed. Prentice Hall PTR, 1995.

[21] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, and L. Zhang, "A framework for end-to-end qos combining rsvp/intserv and differentiated services," IETF, Internet Draft, 1998.

[22] H. Xiao, W. K. G. Seah, A. Lo, and K. Chua, "A flexible quality of service model for mobile ad-hoc networks," in *Proc. of the IEEE 51st Vehicular Technology Conference (VTC)*, 2000, pp. 445–449.

[23] E. M. Royer and C. E. Perkins, "Quality of service for ad hoc on demand distance vector (aodv) routing," IETF MANET, Internet Draft, 2000.

[24] S. Chen and K. Nahrstedt, "Distributed quality-of-service in ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, 1999.

[25] S. J. Lee, G. S. Ahn, and A. T. Campbell, "Improving udp and tcp performance in mobile ad hoc networks with insignia," *IEEE Communications Magazione*, pp. 156–165, 2001.

[26] Bluetooth SIG, *Specification of the Bluetooth System*, Bluetooth Special Interesting Group Std., 2001.

[27] T. S. Rappaport, *Wireless Communications - Principles and Practice*. Prentice Hall PTR, 1996.

[28] C. Zhu and M. S. Corson, "Qos routing for mobile ad hoc networks," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, 2002, pp. 958–967.

[29] 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 Computing*, vol. 1, no. 3, pp. 192–207, 2002.

[30] Z. J. Hass, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proceedings of the IEEE Conference on Computer Communication*, vol. 3, New York, USA, 2002, pp. 1707–1716.

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

[32] 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 *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998.

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