A Pheromone-Aided Multipath QoS Routing Protocol and its Applications in MANETs

Paul Barom Jeon and George Kesidis

Original scientific paper

***Abstract*— In this paper, we present an ant-based multipath QoS routing protocol that utilizes a single link metric combining multiple weighted criteria. The metric is applied to the proposed energy efficient multipath algorithm that considers *both* energy and latency. Energy efficiency is an important issue in mobile ad hoc networks (MANETs) since node energy supplies are stored in batteries. In order to increase the network lifetime it is important to maximize the minimum node energy along a path. As the network topology changes, failures may occur on active routes, resulting in the need for new route discoveries if only single routes per flow are maintained. Frequent new route discovery would, however, increase routing overhead and increase mean and peak packet latency. Using multiple routes simultaneously per flow can be a solution to these problems. Also, a special case of the multipath QoS routing protocol that considers throughput is applied to a security context. A compromised node can obstruct network communication by simply dropping packets that are supposed to be forwarded. In our approach, messages are distributed over multiple paths between source and destination using ant-based QoS routing. In proportion to the throughput of each path, a pheromone-aided routing table is updated and, subsequently, paths that contain malicious nodes are naturally avoided.**

***Index Terms*— QoS Routing, Multipath Routing, Ad Hoc Network, Mobile Network, Wireless Network, Network Protocol.**

I. Introduction

Rapid changes in the topology of mobile ad hoc networks (MANETs) lead to staleness of route information that, in turn, may result in poor routing decisions and a squandering of the overhead associated with routing. In the absence of a viable route, route discovery latency would have a direct impact on packet and flow-level quality-of-service (QoS). Such QoS degradation may not result if multiple routes from the source to the destination are simultaneously maintained, i.e., the network could switch immediately to a "backup" route when an active one fails. This will augment the reliability of data transmission. Multipath routing is also used when load balancing is required. Load balancing is an important field of MANETs because of limited bandwidth and energy restrictions. The sender can utilize multiple paths to satisfy the required bandwidth collectively. To increase the network lifetime, data packets are sent through different multiple paths.

This paper focuses on multipath routing protocols that are *both* timely and energy-efficient for nodes of MANETs accommodating multiple traffic classes. MANETs are often required to be able to operate under "volatile" conditions. Examples of sources of network volatility include finite energy supply (batteries), communication traffic, node mobility, changing environmental conditions affecting the channel and terrain, and enemy activity targeting the network itself. Consequently, there are large existing literatures on single path routing for MANETs, e.g., [26], [42], [43]. In addition, several multipath routing protocols [32], [35], [64] have been proposed to overcome the routing overhead problem of single path routing protocols in a volatile networking context.

For packets that are not delay-sensitive, a routing protocol may choose paths that are rich in energy. The "bottleneck" energy of a path is roughly defined to be the minimum energy supply among its associated nodes. So, the routing protocol could seek to determine the maximum bottleneck ("max-min") energy path to a given destination. Again, this could be done using either a link-state or distance-vector approach. "Energy-aware" routing protocols, taking into account node energy in their link metrics, were proposed previously in, e.g., [9], [50], [63]. In [15], [46], [54], routes with the highest amount of residual energy are chosen to maximize the network's "lifetime" under the basic assumption, of course, that node energy supplies are stored in batteries that cannot be easily, or can never be, replenished. Clearly, energy issues should be considered for latency-critical packets as well as it would be unwise to send them on paths with any component nodes that do not have sufficient energy to transmit them.

In this paper, we propose multipath QoS routing algorithms that jointly manage *both* delay and energy concerns. The principle challenges of such protocols is to be able to route in a highly volatile topology with minimum overhead. We will review the relatively small existing literatures on dual-purpose routing protocols [16], [29], [38].

Multipath routing [6] gives "path diversity" so that communication is less vulnerable to a localized attack, e.g., a relaying node that is either passively eavesdropping or maliciously dropping/modifying data or control packets. Also, judicious replication of important (high-demand) data together with multipath anycasting can make a content distribution network robust against DoS attacks targeting individual peers. Finally, security issues have motivated the use of routing proxies or indirect addressing [55] that may naturally lead to robust

Manuscript received January 10, 2006 and revised May 04, 2006. This paper was presented in part at the 43th Annual Allerton Conference on Communication, Control and Computing, Urbana, IL, 2005, and the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, Montreal, QC, Canada, 2005.

Authors are with the Pennsylvania State University, Computer Science & Engineering and Electrical Engineering, University Park, PA, 16802, USA (e-mail: pbj106@psu.edu; kesidis@engr.psu.edu)

1845-6421/05/5178 © 2006 CCIS

multipath routing.

Since most of the conventional routing protocols in ad hoc wireless networks assume nodes are trustworthy and cooperative, even a small number of compromised nodes can obstruct network communication by discreetly dropping packets [27]. Several schemes [37], [44] have been proposed to detect malicious packet dropping under connection-oriented transport layer protocols, such as TCP. However, TCP is well known not to perform well in ad hoc wireless networks for various reasons such as misinterpretation of packet loss, frequent packet breaks, effects of path length, misinterpretation of congestion window, etc. [39]. Just et al. [27] proposed a probing technique that can detect malicious packet dropping under both connection-oriented and connectionless transport layer protocols.

In the security application of the proposed multipath QoS routing, we focus on avoiding malicious packet dropping, a class of DoS attack. If a node is already compromised at the routing phase, it is assumed to cooperate with other nodes to build legitimate routes between flow sources and destinations. Therefore, a source will not know if the path is insecure at first. At the data transmitting phase, the malicious node drops/modifies packets discreetly to evade detection. If the multiple paths are assumed to be node-disjoint [34], then a system of packet broadcast (replication) on all paths will be robust against such nodes, but this will, of course, generate a significantly large additional network load. So, we propose an ant-based multipath routing protocol that can adaptively avoid malicious packet dropping without additional packet duplication. Once a certain number, N, of packets are delivered through a given path, the destination generates a backward ant (BANT) [18] packet and sends it back along that path to source. After receiving the BANT packet, the source node updates its related pheromone-aided routing table. With the aid of pheromone mechanisms, the path that contains a malicious node is naturally avoided in a computationally and bandwidth efficient fashion.

This paper is organized as follows. In Section II, we overview a number of previous protocols that are related to our work. In section III, routing overhead under various multiple node-disjoint paths algorithms is discussed. We propose in section IV a more adaptive routing protocol to accommodate *both* energy and delay metrics under volatile conditions. In section V, we provide a simulation study of our proposed multipath QoS routing protocol. In section VI, the proposed multipath QoS routing protocol is applied to a security context in order to adaptively avoid malicious packet dropping. Finally, conclusions are drawn in Section VII.

II. Related Work

A. Multipath Routing

Prominent routing protocols for wireless MANETs, such as AODV [43] and DSR [26], usually develop a single path between source and destination for a given flow. When the developed route fails, the sender would need to discover a new route to the receiver. This new route discovery will result in additional packet delay and network overhead. If multiple paths were developed in the first stage, when the first active path (possibly representing the shortest delay path) fails, a new path to the receiver can be immediately used.

Split Multipath Routing (SMR) [32] protocol is an extension of DSR, inheriting the on-demand, source routing characteristics of DSR. SMR builds maximally disjoint paths. SMR discovers two paths that share as few nodes and links as possible. One of the two routes is the shortest delay path. Maximally disjoint paths are preferred to prevent congestion on shared nodes/links, to accomplish efficient utilization of the network resources, and for fault tolerance. Initially, the source floods ROUTE REQUEST (RREQ) packets to the entire network to discover routes to the destination. Unlike DSR, the intermediate nodes do not maintain a route cache and, consequently, do not respond to the RREQs. Not every duplicate RREQs are discarded at intermediate nodes. Intermediate nodes forward the duplicate RREQ if its incoming link is different and its hopcount is not larger than that of previous received RREQs. As a result of flooding, several RREQs which traveled through different paths arrive at the destination. The shortest delay route is determined when the first RREQ arrives at the destination. After sending ROUTE REPLY (RREP) back to the source, the destination waits for certain amount of time to receive more RREQs and learn more possible routes to the source. Among the possible paths, it selects two mostly disjoint paths. On discovering the first route, it is used to send data packets to the destination. When more than one route is discovered, data traffic is distributed into multiple paths using a per-packet allocation scheme.

AOMDV [35] is a multipath version of the well-known on-demand single path routing protocol AODV. It discovers loop-free and link-disjoint multiple paths between source and destination. By flooding the RREQs from the source to the destination multiple reverse paths are discovered both at the intermediate nodes and the destination node. Multiple RREPs are sent from the destination to the source. The RREPs travel back to the source through the previously established multiple reverse paths and form multiple forward paths to the destination at the source and intermediate nodes. The AOMDV protocol has two essential elements: establishing and maintaining multiple loop-free paths and finding link-disjoint paths. To establish loop-free paths, each node i maintains an "advertised hopcount" defined as the "maximum" hopcount of the multiple paths, for a destination d. Each node will accept an alternate route to the destination only if the hopcount of the alternate route is smaller than the advertised hopcount. Since the maximum hopcount is considered, the advertised hopcount will remain unchanged for the same sequence number. The advertised hopcount will be reset when node i receives a route advertisement for destination d with a greater sequence number. In order to discover multiple link-disjoint paths, each RREQ contains a "firsthop" field, defined as the neighbor of the source which RREQ has traveled at the first hop. In addition, "firsthop_list" for each RREQ is kept at each node. An intermediate node does not discard duplicate copies of RREQ immediately. On receiving a duplicate RREQ, the intermediate node checks whether it can form a new node-disjoint path to the source using "firsthop" and "firsthop_list".

Unlike intermediate nodes, the destination sends back several RREPs to the source, through unique neighbor of destination, regardless of the firsthop of the RREQ. Link-disjointness is assured at the first hop of the RREP by using different neighbors. After the first hop, the RREP follows the already established node-disjoint and thus link-disjoint reverse paths to the source.

AODVM [64] is also an extension of AODV. It discovers multiple node-disjoint paths. Instead of discarding all the duplicate RREQs, intermediate nodes keep information of each RREQ in the so called RREQ table. Also, intermediate nodes are not permitted to send RREPs directly to the source. Whenever the destination receives a RREQ it updates the sequence number and sends back a RREP to the source through the path which RREQ has traveled. When the RREP arrives at an intermediate node, a forward path from the intermediate node to the destination node is set up. Subsequently, the intermediate node selects a shortest path to the source from its RREQ table and sends the RREP via the path. In order to ensure the node-disjointness when nodes overhear any neighboring node sending a RREP packet, they delete the transmitting neighbor node from its RREQ tables.

Under multipath routing, a basic question is how to chose a path when multiple paths are available at a certain node to a destination. AOMDV simply selects the path that was discovered first, and SMR uses a per-packet allocation scheme. When there are more than one available path to a destination, SMR allocates these multiple paths in turn to a packet to be delivered to the destination. These simple schemes, however, will not fit in well with a multipriority paradigm, since the different priorities cannot be taken into consideration when selecting a path. In the following, we consider multipath and multipriority issues jointly when discovering routes to the destination. Instead of developing an optimal path for each priority, a suboptimal path is chosen among the multiple paths considering the priority. The proposed new method has low overhead since it does not execute multiple routing algorithms to satisfy the requirement of different service priorities. With the aid of pheromone mechanisms, low computational complexity per node is achieved. A source node determines the next hop to a destination based only on the information delivered by the pheromone, hence routing is highly decentralized.

B. QoS Routing

In [12], the author proposed to maximize network lifetime by traffic flow augmentation and flow redirection to balance the energy consumption rates among the nodes in proportion to their energy reserves. In particular, they showed that delay-constrained routing was NP-complete. However in [13], the author proposed a heuristic algorithm to first reduce the NP-complete problem to a simpler one that could be solved in polynomial time, but the proposed solution was mainly for QoS routing in a multimedia network and energy issues could not be treated as a QoS parameter. Moreover, the time and computational complexity of the algorithms might not be ideal for ad hoc networks.

Taking inspiration from this, Feng and Douligeris [16] proposed a single weighted link metric combining both delay and cost considerations. However, this algorithm requires that the costs to be additive. For the problem at hand, delay is a cumulative parameter but the minimum energy resources should not be added along a route. Also, there is an assumption that both of the costs are of comparable magnitudes, which raises an issue of appropriate choice of dimensions.

A possible solution to route optimization while dealing with additive and non-additive metrics was proposed in [29]. However, this approach combines the two attributes to make routing decisions. In this way we lose control of the compromise with the energy for better delay performance. In other words, a route with lower cumulative delay may occasionally be selected even though the route is energy-poor. As a result of this, a node may at times only find paths with large delays as the network tries to preserve energy-poor paths.

Note that two independent routing algorithms running in parallel, one for each link metric, have the disadvantage of requiring twice the overhead as a single algorithm addressing both metrics. One could envision a synchronization of the two protocols so that a single packet would be required to update both metrics; however, both metrics would not necessarily require updates at the same times nor with the same frequency and, therefore, a single-packet update for both metrics may not result in substantial overhead savings. The routing information could also be piggybacked on new/additional data packet headers, but this has the disadvantage of reducing general transmission efficiency (header-to-packet ratio) of the network and may add latency to the routing algorithm as one would want to wait for a priority packet to piggyback. Typically, low-volume routing messaging would be treated as priority communication.

C. Ant Routing

Ants in routing algorithm are simple agents wandering over the network, from one node to another, leaving trails by depositing *pheromones* on the nodes they visit in a manner that will be clarified shortly. Ant-based routing (AntNeT [10]) for a group of mobile nodes is an adaptive and distributed approach inspired by an ant-colony metaphor involving the notion of pheromone levels. Many variations of ant routing have been proposed for routing problems in various communication-network contexts [23], [33], [49], [58], [65], [66]. For MANETs, ant-based algorithms can cope with highly dynamic topologies and link/channel qualities resulting in multiple, but fleeting, paths [18]. Ant-based routing algorithms have several advantages [4]: routing determination using only local information, inherently scalable, and robust and responsive to environment change.

AntNet consists of two types of homogeneous mobile agents (each embodied in a single packet) respectively called *forward* and *backward* ants. From every node s, a forward ant (FANT) is launched periodically to a randomly selected destination node d. At each node k visited by a FANT, the next hop node is determined by a random forwarding decision. If pheromone information is available, the probability P_{jd} of forwarding to a given neighbor j of k ($j \in S_k$[1]) is proportional to the current

[1] A set of neighboring nodes of node i is denoted by S_i.

amount of pheromone at k that is jointly associated with d and j.

Initially, all pheromone levels could be equal (or, in the absence of pheromone to a destination, a purely random forwarding decision is made) but they will vary with time as described in the following. A FANT is converted to a backward ant (BANT) when it reaches d. The BANT follows the same path in the opposite direction as that of its corresponding forward ant. When the backward ant is transmitted to node k by node f ($k \in S_f$), the forwarding probability P_{fd} at k is increased and P_{dn} is decreased for all $n \in S_k, n \neq f$. Once the backward ant returns to its source, s, the (one-way) trip time experienced by the FANT is measured and the pheromone levels associated with d at s are adjusted accordingly.

The Ant-Colony-Based Routing Algorithm (ARA) [18], suitable for MANETs, is based both on "swarm intelligence" and ant-colony meta-heuristics. ARA consists of three phases: route discovery, route maintenance, and route failure handling. In the route discovery phase, new routes between nodes are discovered with the use of a forward-and-backward ants, similar to AntNet. Routes are maintained by subsequent data packets, i.e., as the data traverse the network, node pheromone values are modified so that their paths are "reinforced." Also, as in nature, pheromone values decay with time in the absence of such reinforcement. Routing (link) failures, usually caused by node mobility, are detected through missing acknowledgements. When a node detects a routing error, the pheromone value associated with the "missing link" is set to 0.

Termite [45] is an another ant-based routing algorithm that is similar to ARA. However, unlike the ARA, pheromone is not considered in the route discovery phase. Instead of the forward and backward ants, route request (RREQ) and route reply (RREP) control packets are used to discover the routes. The RREQ packet randomly walks, not floods, through the network to discover a route to the destination. Pheromone levels are used for routing data packets and proactive "seed" packets are introduced for route maintenance.

Both energy and delay issues were considered in [38]. Only delay quantities, however, are considered when computing the pheromone values and forwarding probabilities. The dissipated energy of a node after each ant passes through it is calculated by

$$\Delta E_{ij} = \frac{K}{(D_{ij})^2} \tag{1}$$

where K is the amount of energy to transmit the ant over a single unit distance, and D_{ij} is the Euclidean distance between node i and j [36]. The residual node energy at time t is computed by:

$$E_i(t) = E_i(t-1) - \sum_j \Delta E_{ij}. \tag{2}$$

When a node's energy level simply drops below a pre-specified threshold value, the node is removed from the sensor network and alternative routes are found.

D. Secure Routing

In ad hoc wireless networks, intrusions can be classified into two categories: *passive* and *active* attacks. A passive intrusion does not disrupt the functioning of the network; instead, the attacker eavesdrops on the traffic flowing across the network to discover valuable information without modifying the data. Since a passive attack does not affect the functioning of the network, it is very difficult to detect. Encryption schemes are usually used to protect the data from an intruder/attacker. Unlike a passive attack, an active attack modifies or drops messages thereby obstructing the functioning of the network. Messages include both routing control packets and data packets. An adversary can attack routing packets resulting in an inefficient routing table at the source. On the other hand, an adversary can attack data packets resulting in incomplete transmission, even though it cooperates with other nodes to build legitimate routes between sources and destinations. Examples of active attacks and proposed solutions are Wormhole attacks (e.g., Packet Leashes [21]), Blackhole attacks (e.g., [14]), Byzantine attacks (e.g., [5]), and routing attacks (e.g., [22], SEAD [19], ARAN [48], ARIADNE [20]).

In a security context, multipath routing has several advantages including the avoidance of passive attacks, the avoidance of malicious packet dropping, and the robustness against focused DoS attacks. The avoidance of passive attacks (e.g., eavesdropping) can be achieved by simply dispersing the traffic over the existing multiple paths between source and destination so an attacker can only successfully intercept a portion of the transmitted message. In addition to this basic idea, various schemes have been added in order to increase confidentiality [7], [8], [41]. If there exist k paths between source and destination and n compromised intermediate nodes, although an attacker successfully drops part of the transmitted message, the original message can be safely delivered to the destination (unless n is greater than k and at least one node in each path are compromised). Note the tactic of multipath routing can be used in concert with intrusion detection systems [27], [37], [44] that are designed to detect intruders and then respond by, e.g., avoiding paths in which they reside. If an intruder attempts to substitute fraudulent packets, one employ a traceback mechanism to identify the intruders, e.g., trustworthy nodes could *log* packets at various points throughout the network and then using some extraction ("data mining") techniques to find the path fraudulent packets traversed, see [47]. Snoeren et al. [51], [52] proposed a modification to this approach, called the Source Path Isolation Engine (SPIE), that hashes and stores only the first 28 bytes of a packet, thereby avoiding 99.9% of all collisions (false positives due to many-to-one mapping of the hash) while saving tremendous amounts of storage space. Alternatively, one could mark packets for traceback [53]. Finally, robustness against focused DoS attacks can also be achieved by transmitting replicated packets over multiple paths [30]. Although some portion of the intermediate nodes may be attacked and rendered unable to forward messages, with the help of redundant messages, the original message can be safely delivered to the destination.

III. Routing Overhead Analysis

In an on-demand protocol, when a node needs to discover a route to a destination, it broadcasts route request (RREQ) packets into the network. In order to reduce the routing overhead, each intermediate node typically forwards only one RREQ originating from the same source node. Most of the existing on-demand single path routing protocols, such as AODV [43], DSR [26], and LAR [28], only forward the first arrived RREQ. Consider the simple network shown in Figure 1. Suppose that a RREQ that followed the path $S \rightarrow a \rightarrow b \rightarrow e \rightarrow f$ arrived first at node f. When another RREQ that followed the path $S \rightarrow i \rightarrow j \rightarrow k \rightarrow f$ arrives at node f, it will be dropped since node f already has seen a RREQ that originated from the same source S. Such a RREQ forwarding method limits the possibility of finding feasible multiple routes, e.g., only the route $S \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow D$ is discovered although there exits another feasible route $S \rightarrow i \rightarrow j \rightarrow k \rightarrow f \rightarrow g \rightarrow h \rightarrow D$.

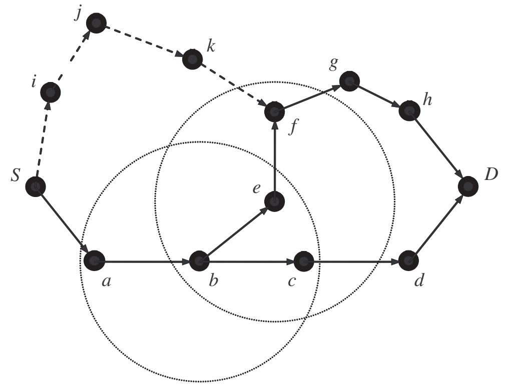


Fig. 1. Example network topology

Unlike single path routing protocols, multipath routing protocols do not always forward only one RREQ. In Split Multipath Routing (SMR) [32], not all duplicate RREQs are discarded at intermediate nodes. Intermediate nodes forward the duplicate RREQ if its incoming link is different and its hopcount is not larger than that of previous received RREQs. Therefore, node f in Figure 1 forwards *both* RREQs from node k and e. In SMR, the destination node D discovers two node-disjoint paths from the *three* received RREQs.

In AOMDV [35], an intermediate node does not discard duplicate copies of a RREQ immediately. On receiving a duplicate RREQ, the intermediate node checks whether it can form a new node-disjoint path to the source using *firsthop*, *firsthop_list*, and $advertised\_hopcount$, see [35] for detail definitions. On receiving a route reply (RREP) packet sent from the destination, an intermediate node selects one reverse path from the early stored reverse paths. Similar to AODV, an intermediate node only forwards the first arriving RREQ in AOMDV. For the example network shown in Figure 1, node f does not set up a reverse path to node k although it receives a RREQ via node k since $advertised\_hopcout_f^k$ is not greater than $advertised\_hopcout_f^e$. Therefore, only one path $S \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow D$ is found.

In AODVM [64], instead of discarding all the duplicate RREQs, intermediate nodes keep information of each RREQ in the so called RREQ table. When the RREP arrives at an intermediate node, a forward path from the intermediate node to the destination node is set up. Subsequently, the intermediate node selects a shortest path to the source from its RREQ table and sends the RREP via the path. In order to ensure the node-disjointness when nodes overhear any neighboring node sending a RREP packet, they delete the transmitting neighboring node from its RREQ tables. Similar to AOMDV, AODVM cannot discover two node-disjoint paths for the example network shown in Figure 1. At node f, the RREP is forwarded to node e since two paths stored in RREQ table of node f have same number of hops to the source and the RREQ via node e arrived first at node f. Just as in AOMDV, only one path $S \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow D$ is found.

In [60], a multipath routing algorithm based on selective broadcast (SB) method was proposed. When an intermediate node receives a RREQ, it caches the RREQ and rebroadcast the RREQ only when the traveled path is node-disjoint with the paths included in previously cached same RREQs. Similar to SMR, node f in Figure 1 rebroadcasts *both* RREQs from node k and e. Just as in SMR, the destination node D discovers two node-disjoint paths from the *three* received RREQs.

The route discovery result of different routing algorithms for the example network topology is summarized in Table I.

TABLE I
NUMBER OF NODE-DISJOINT PATHS VS. NUMBER OF RREQS

	SMR	AOMDV	AODVM	SB
Node-Disjoint Paths	2	1	1	2
Received RREQs at D	3	2	2	3

The result clearly shows that more routing overhead is required to discover more node-disjoint paths. However, the increased RREQs, which are broadcasted at each intermediate node, can degrade the performance of the routing protocol [57], [59]. When an active path gets broken and a source node has no cached alternative path, the source node broadcasts RREQ to discover a route to a destination. However, in a congested network, these broadcasted RREQs are more likely to collide with data packes of other connections.

In order to reduce the routing overhead of the *selective broadcast* method, a *heuristic redirection* method was proposed in [61]. Our proposed multipath routing algorithm adopts heuristic redirection in order to discover multiple paths with less overhead. Details of our route discovery method is described in Section IV-A.

Figure 2 shows the average ratio of the number of discovered node-disjoint paths with 95% confidence interval using two multipath algorithms to the number of hops of the shortest path. The sub-ideal number of node-disjoint paths are discovered using a method similar to that of [64]. The source first discovers a path to the destination using AODV [43]. The nodes on the first found path are then eliminated and AODV is executed again to find another path. Theses steps are repeated until no node-disjoint paths are found between the source and the destination. Details of our simulation environments are described in Section V.

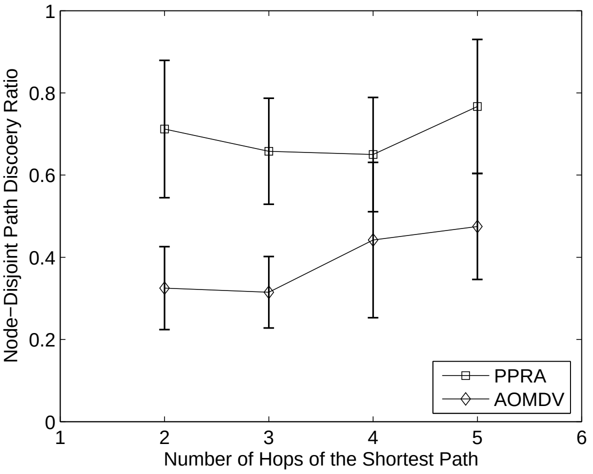


Fig. 2. Ratio of the number of discovered node-disjoint paths to the number of hops of the shortest path.

IV. PPRA: Prioritized Pheromone Aided Routing Algorithm

Rapid changes in network topology leads to staleness of routes resulting in poor routing decisions and a squandering of the overhead associated with routing. This is especially costly when energy supplies are dwindling or successful communication requires heightened transmission energy due to increased channel noise. For such a volatile networking context, a highly distributed, quickly responsive (adaptive) and bandwidth-efficient multipath routing algorithm is required.

A. Route Discovery

Conventional ant routing algorithms require significant overhead for preliminary route set-up [17] possibly resulting in slow route convergence which is not satisfactory for dynamic ad hoc networks. A convergence result for pheromone load balancing, resolving the optimal path(s) among pairs of nodes, was given by us in [24]. Formal convergence results for ant routing were first announced in [31], in particular Theorem 2 of [31] for "regular" ants (in the terminology of [56]). In order to overcome the slow route convergence problem with less routing overhead, the route discovery characteristic of our scheme is inherited from the heuristic redirection method [62].

When a traffic source does not have an active route to a destination, the source initiates route discovery by broadcasting a forward ant (FANT) packet. The source address is included in the FANT, and as it propagates, the addresses of intermediate nodes that it visits are appended prior to forwarding. Unlike other single path routing algorithms, in order to discover multiple paths, intermediate nodes do not discard duplicate FANTs, instead the latter received FANTs are cached at the intermediate nodes. When a FANT reaches the destination node, the destination generates a backward ant (BANT) packet for the source node. Since the destination has a route to the source contained in the FANT it received, it does not flood the BANT but forwards it to one of the neighboring nodes if the path contained in the received FANT is node-disjoint with all cached paths of previously received FANTs. The intermediate nodes heuristically redirect the received BANTs originated from the destination node in order to increase the chance of finding a new node-disjoint path. More detail of the heuristic redirection method is well described in [62].

B. Energy Efficient Source Routing

One of the main problem of source routing is that the packet length is proportional to the path length. Therefore, the energy consumption also increases proportionally. In order to overcome this problem, an energy efficient method is added to [25] as following.

At the source and intermediate nodes, PPRA checks if the selected path has been used before. If the path has not been used before, full route information is added to the data packet similar to the conventional source routing protocols. Before forwarding the data packet, PPRA builds a routing table using the destination and next hop information extracted from the route information added to the data packet. When the first data packet reaches the destination node, proper routing tables are built at the corresponding source and all intermediate nodes. When the next data packet is to be sent, PPRA notices that the selected path is used before and does not add any additional route information to the data packet. Instead, PPRA utilizes the routing table built upon first use of the path.

By using such a method, the additional route information does not have to be always added to the data packet. This will reduce the total packet size resulting in a less energy consumption.

When a node is responsible for forwarding multiple traffic flows, energy consumption of the node increases proportionally. In order to overcome this problem, a path with no node responsible for multiple traffic flows is preferred. At the route discovery phase, the number of traffic flows that utilize the node is collected and sent to the source. The number of traffic flows of each node is examined and the maximum is cached as max_node_used. The max_node_used is used when selecting a path to be cached. If two paths have same path length, the path with smaller max_node_used is preferred. Note that maintaining more than two alternate paths usually does not affect the performance considerably [40]. For this reason, we maintain only two multiple paths.

C. Route Maintenance

The source nodes maintain a routing table that contains entries of neighboring nodes to reach destination nodes. For example, in Figure 3, if node Z forwards a BANT from node W to node X, node X creates a pheromone state entry for reaching node W through node Z. An example routing table is shown in Table II. When the source receives the BANT, it has an entry for reaching the destination through one of its neighbors. Since duplicate FANTs are not discarded, the destination node may send multiple BANTs back to the source. Note that we specifically consider node-disjoint paths for simulation simplicity [62].

FANTs are periodically sent from the source to the destination, after sending every N data packets along the selected

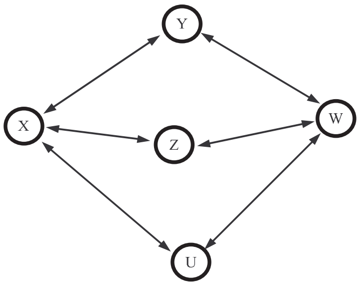


Fig. 3. Small example network topology

TABLE II
ROUTING TABLE AT NODE X FOR CASE 1

Destination Node	Neighbor Node	TTL -Pheromone	Energy -Pheromone
W	Y	$\delta^x(w,y)$	$e^x(w,y)$
W	Z	$\delta^x(w,z)$	$e^x(w,z)$
W	U	$\delta^x(w,u)$	$e^x(w,u)$

path. Once the destination receives the FANT, it sends a BANT back to the source using the same path the FANT has traveled. Therefore, both energy and delay information of the path are collected and delivered to the source. The pheromone levels of the source are updated using equations (3) and (5). Similar method to maintain the route periodically was introduced in [11].

D. Routing using Energy and Delay Metrics

The ant routing framework has two types of feedback: positive feedback increases the pheromone levels on routes actively carrying ant packets and negative feedback periodically decreases pheromone values to limit the effects of stale information. Routing decisions tend to favor paths with higher pheromone levels and, when allowed to converge, shortest end-to-end paths are empirically observed to be favored [18]. In the following, we propose a modified ant mechanism algorithm that uses energy and delay metrics to perform updates of pheromone levels. Assuming a control packet containing both energy and delay information, a separate pheromone level will be maintained for each traffic type.

The time-to-live (TTL) field for ant-like routing provides explicit *distance* (hopcount) information to a packet's source. TTL (hopcount) information is widely used in existing routing mechanisms, see, e.g., [43]. We will exploit a "TTL pheromone" based on TTL data to improve the performance of an ant-like routing algorithm.

Two routing algorithms are developed. In the first algorithm, ant packet headers are assumed to have two fields used for routing: one to indicate bottleneck residual energy of a path and the other being TTL. This data will be used to maintain both bottleneck residual energy and TTL pheromone levels in the nodes. More specifically, if the residual energy of a forwarding node is lower than the bottleneck residual energy inscribed in the packet, the node overwrites the packet's energy field with its own residual energy level. Latency-critical traffic will be routed based on *both* pheromones whereas latency-non-critical packets will use only energy pheromone levels. Note that we continue to assume that latency-critical and latency-non-critical packets are separately queued at each node.

In the second algorithm, packet headers have fields that:

- track the minimum residual energy of the nodes that relay them (as in the first algorithm) and
- track the cumulative delay based on backlog information of queued packets destined to the packet's source.

So, when a packet reaches its destination, it contains the minimum residual energy and the cumulative queuing delay of its route back to its destination. Thus, energy and delay (instead of TTL) pheromone levels will be maintained at each node.

1) TTL-pheromone and Energy-pheromone: Consider Figure 3. Upon receipt of a new ant packet transmitted from node W to node X via Z, with TTL value $TTL(w,z)^2$, node X calculates the TTL pheromone value $\delta^x(w,z)$ *to* node W through node Z using:

$$\delta^x(w,z) = \beta_1 TTL(w,z) + \delta^x(w,z) \tag{3}$$

where β_1 is a scaling constant.

Node X maintains a routing table, as shown in Table II, containing both a TTL-pheromone δ and energy-pheromone e levels. Pheromone values will decrease in time in the absence of positive feedback. The periodic decay for the TTL-pheromone is governed by the iteration:

$$\delta^x(w,z) = \delta^x(w,z)\beta_2 \tag{4}$$

where $0 < \beta_2 < 1$.

For a highly volatile network, we can increase the frequency of decay or decrease the β_2 value. This ensures that stale routes decay faster.

Similar to equation (3), the energy-pheromone is calculated as:

$$e^x(w,z) = \alpha_1 E_{min}(w,z) + e^x(w,z). \tag{5}$$

where α_1 is a scaling constant. Periodic decay for energy-pheromone levels is governed by:

$$e^x(w,z) = e^x(w,z)\alpha_2 \tag{6}$$

where α_2 is a constant satisfying $0 < \alpha_2 < 1$. Again for highly volatile networks, periodic decays are performed more frequently to ensure faster decay of stale route information.

The latency-non-critical packets are routed according to the probabilities that are simply proportional to energy-pheromone levels. Specifically, latency-non-critical packets at X destined to node W are routed through Z with probability:

$$p_e^x(w,z) = \frac{e^x(w,z)}{e^x(w,z) + e^x(w,y) + e^x(w,u)}. \tag{7}$$

For latency-critical packets, first define a decision probability proportional to the TTL-pheromone:

$$p_\delta^x(w,z) = \frac{\delta^x(w,z)}{\delta^x(w,z)\ + \delta^x(w,y)\ + \delta^x(w,u)}. \tag{8}$$

[2] Note that the TTL value is decremented as packet propagates the network.

Routing of latency-critical packets should consider *both* energy and delay pheromone levels. We will use the following decision probability to route through Z packets at X destined to W:

$$p^x_{lat}(w,z) = \frac{p^x_\delta(w,z) \ + \ \theta p^x_e(w,z)}{\sum_{i\in\{z,y,u\}}[\ p^x_\delta(w,i) \ + \ \theta p^x_e(w,i)\ ]} \quad (9)$$

where $0 < \theta < 1$.

Note that $p^x_{lat}(w,z)$ combines *both* energy-pheromone and TTL-pheromone values into one quantity with a comparable magnitude, which is the main contribution of this paper. By normalizing the pheromones, we can make both pheromones have the same dimension. It is similar to the combination of delay and cost metric in QoS routing [16], but unlike [16] the combination is of two *similar* quantities as the TTL and Energy pheromones are *normalized* values. Also, $p^x_{lat}(w,z)$ considers *both* additive and non-additive metrics whereas [16] considers *only* additive metrics.

2) Delay-pheromone and Energy-pheromone: This second algorithm is very similar to the previous one, the only difference is that a delay-pheromone replaces the TTL-pheromone in the routing tables. More specifically, upon receipt of an ant packet at X from W via Z carrying cumulative delay (back to W) metric $D(w,z)$, the delay-pheromone is updated as

$$d^x(w,z) = \gamma_1 D(w,z) + d^x(w,z) \quad (10)$$

where γ_1 is a scaling constant. The periodic decay equation is:

$$d^x(w,z) = d^x(w,z)\gamma_2 \quad (11)$$

with $\gamma_2 > 1$. The component $p^x_d(w,z)$, that replaces $p^x_\delta(w,z)$, of the routing decision probabilities (9) for latency-critical packets that is based on delay-pheromone is computed as shown in (12).

$$p^x_d(w,z) = \frac{1/d^x(w,z)}{1/d^x(w,z) + 1/d^x(w,y) + 1/d^x(w,u)} \quad (12)$$

V. Performance of PPRA

We have simulated PPRA using ns-2 [1] to validate its efficiency and ability under volatile MANETs environments. Packet delivery ratio, end-to-end delay, routing overhead, route discovery overhead, and residual node energy were used as metrics to compare the performance of PPRA with AOMDV and AODV routing protocols. Each simulation result (each reported point on each curve) represents an average of 10 independent trials.

Unlike the results of [24], the simulation results did not converge to a single optimal path. As the minimum residual energy of a node along a path varied with time, the amount of delivered pheromone changed. Therefore, instead of converging to a single path, the multiple paths were used alternatively. Most of the time, as most of the traffic was assumed to be low priority traffic, the source tried to use energy-rich path, which was determined by equation (9) and changed with time. Hence, load balancing was achieved and the network lifetime was maximized.

A. Simulation Environment

In our simulation study, we assumed a 1000m x 1000m area, where fifty nodes were disbursed randomly with the communication range of a node being 250 meters. The nodes employed the IEEE 802.11b medium access mechanism with 2 Mbps channel bit rate. To examine the effect of the routing algorithm only, the network was moderately loaded and ten source-destination connection pair was considered. A constant bit rate (CBR) traffic source model was used to generate data traffic of 1024 Byte packets. Using the CBR model, the source sent one data packet per second to the destination on average. Among all of the data, 10% was designated latency-critical. Latency-critical type had priority over latency-non-critical type and whether a packet was of the latency-critical or latency-non-critical type was determined at random and independently. In the simulation, we assumed that nodes had 20.0 Joules of battery energy initially. The transmitting and receiving power required by a node was set to 1.3272 W and 0.96696 W respectively [3].

Each simulation was run for 900 seconds using random waypoint mobility model with maximum speed of 20 m/sec. Seven different pause times, from 0 to 900, were used to observe the effect of the mobility. The pheromone values of each path were updated every 10 data packets. At this update phase, BANT packets, which carried the energy and delay information of the path, were used. The destination node sent back a BANT packet to the source along a path after receiving 10 data packets that traveled along the path.

B. Performance Metrics

Five metrics were taken into consideration: *Packet delivery ratio* is the ratio of successfully delivered data packets to the total data packets sent from the source to the destination. *End-to-end delay* is the amount of time needed to successfully deliver a packet from the source to the destination. End-to-end delays were observed separately for both priority (latency-critical) packets and nonpriority (latency-non-critical) packets. *Routing overhead* is the ratio of routing packets transmitted to the total data packets delivered. Routing packets include control packets used for route discovery, route maintenance, and pheromone updates. *Route discovery overhead* is the total number of route discovery phases and the total number of route request (FANT) packets distributed over the entire network. Finally, *node energy* is the average node energy and minimum node energy that were observed.

C. Simulation Results

Figures 4 to 10 shows the simulation results under different pause times. One data packet was generated every second ($\lambda = 1$) and seven different pause times were considered. The maximum speed was set to 20 m/sec. Each simulation result for PPRA was compared to that of AOMDV and AODV.

Figure 4 shows the packet delivery ratio (PDR) of PPRA, AOMDV, and AODV. The PDR tends to increase as the pause time increases. This is manifest since the active path is less likely to break as the network becomes static. However,

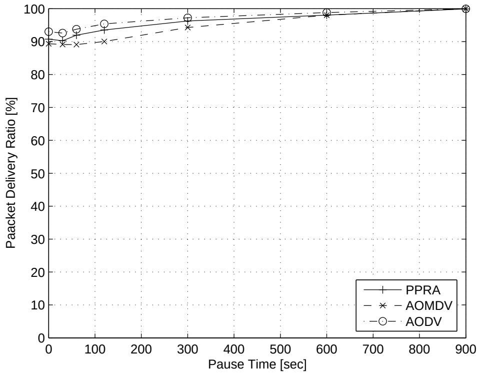


Fig. 4. Packet Delivery Ratio ($\lambda = 1$)

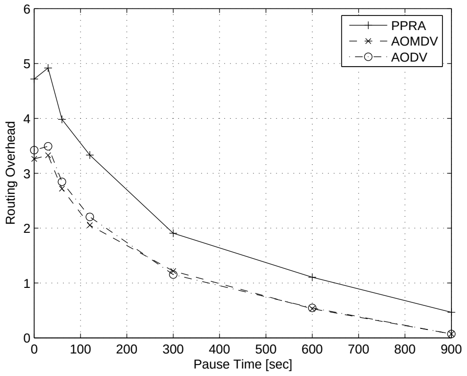


Fig. 6. Routing Overhead ($\lambda = 1$)

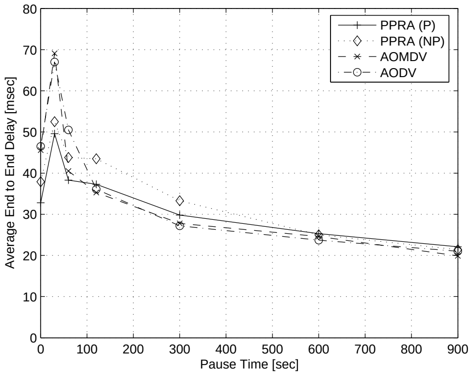


Fig. 5. End-to-End Delay ($\lambda = 1$)

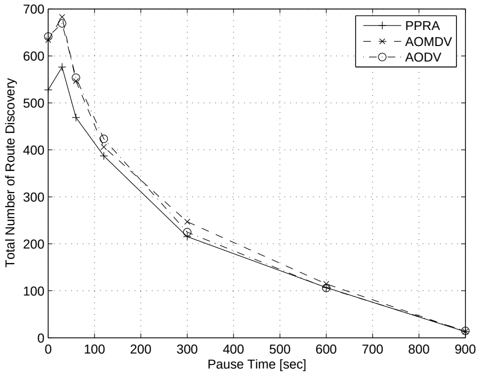


Fig. 7. Total Number of Route Discovery ($\lambda = 1$)

the PDR first decreases as the pause time increases. Due to mobility, the active path may break. When all paths, including the backup paths, to the destination break, a new path can be discovered only after the change of topology of the network, i.e. a node that can form a path to the destination should come into the transmission range. Note that the change of topology is proportional to mobility. Hence, as mobility decreases it becomes more difficult to recover from the broken path. This reason explains the valley that appears when the pause time is 30 seconds in Figure 4. Note that AODV shows better PDR, especially under high mobility. Under high mobility, both the active path and backup path are more likely to break. Therefore, the packet sent out over the backup path just after the break of the active path is more likely to be dropped under high mobility.

In Figure 5, we can see that the end-to-end delay of latency-critical packets (P) is reduced by applying PPRA. Since multiple paths were discovered, when a path to the destination breaks, packets could immediately continue to be forwarded using a backup path without a new route discovery. Obviously, this reduced the end-to-end delay. We can observe that, in general, the end-to-end delay of latency-non-critical packets (NP) is greater than that of P packets. Note that AOMDV shows better performance from the end-to-end delay perspective under moderate and low mobility. Since PPRA considers both delay and energy jointly, not all priority packets were sent along the shortest path. For this reason PPRA generally requires more end-to-end delay than AOMDV. Unlike PPRA, AOMDV used the shortest path first. It switched to the alternate path when the shortest path was not available.

The routing overhead is shown in Figure 6. Since more control packets are required at the route discovery phase and extra control packets are required periodically to monitor the condition of the paths, the routing overhead of PPRA is higher than other protocols. The overhead for path monitoring can be reduced by piggybacking the pheromone information on data packets if appropriate traffic exists in opposite direction. Because of the periodic updates, PPRA requires certain amount of routing overhead constantly.

Figure 7 shows the result of number route discovery under various pause times. The number of route discoveries is less than that of the other two protocols, especially under high mobility, since PPRA discovers and maintains multiple routes to the destination. Figure 8 shows the result of total route request (FANT) packets distributed over the entire network. As explained in section III, PPRA requires more number of FANTs to be distributed per route discovery. However, since the number of route discovery is less than other protocols, the total number of distributed FANTs is comparable to that of others.

Figures 9 and 10 depict the average of residual node energy and minimum of residual node energy respectively.

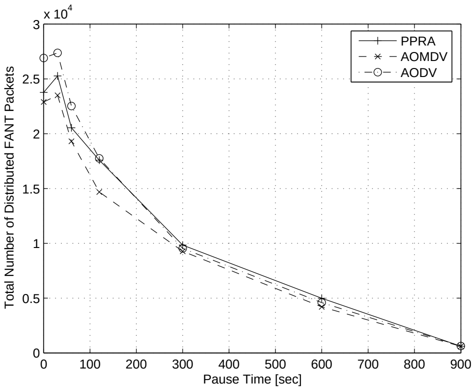


Fig. 8. Total Number of Distributed FANT Packets ($\lambda = 1$)

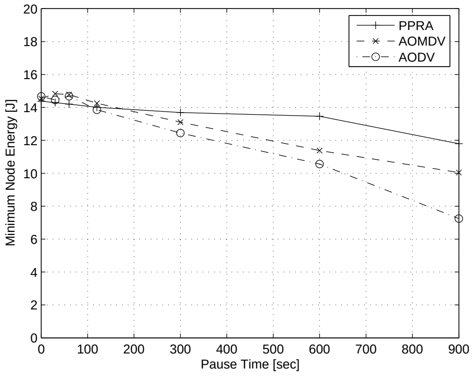


Fig. 10. Average Minimum Node Energy ($\lambda = 1$)

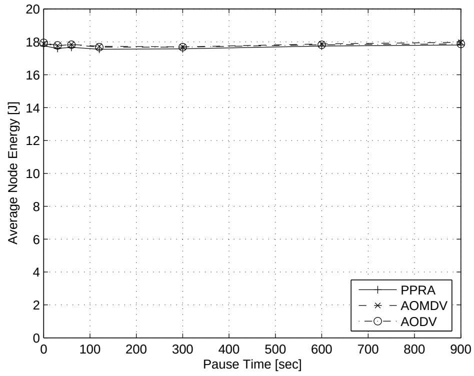


Fig. 9. Average Node Energy ($\lambda = 1$)

Minimum node energy was defined as the residual energy of a node that had the smallest residual energy. Due to periodic pheromone updates, fundamentally, PPRA consumed more energy, in total, than other routing schemes under same circumstances. However, we can see that the average node energies under PPRA are similar to those of other protocols. This is due to the small packet size of the control packet. Since other routing schemes did not consider the priority of the data packets, all latency-critical and latency-non-critical packets were sent using the same path. Once the path was selected, the path was used until it became unavailable. PPRA, however, selected a path depending on the priority of data packets and pheromone values of multiple paths. Different type of data packets used different paths in PPRA. Hence, the discovered multiple routes were alternately used resulting in higher minimum node energy. From the results we can observe that PPRA increases the minimum residual node energy with comparable total energy consumption.

More simulations using different transmission rate were conducted by us, e.g., taking $\lambda = 2$, all leading to similar results as those reported above.

VI. Avoiding Malicious Packet Dropping

The ant routing framework has two types of feedback: positive feedback increases the pheromone levels on routes actively carrying ant packets and negative feedback periodically decreases pheromone values to limit the effects of stale information. Routing decisions tend to favor paths with higher pheromone levels and, when allowed to converge, shortest end-to-end paths are empirically observed to be favored [18].

In the following, we propose a modified ant mechanism algorithm that uses throughput metrics to perform updates of pheromone levels. Control packets (BANTs) are periodically sent from the destination to the source, after receiving every N data packets, along a path that data packets have traveled. Once the source receives a BANT, it increases the pheromone level of the path that the BANT has traveled. Therefore, the pheromone level of a path is proportional to the throughput of the path. If all paths are secure, i.e., there are no malicious packet dropping along the paths, the data packets will eventually use the highest throughput as shown in [24].

Considering Figure 3, upon receipt of a new BANT packet transmitted from node W to node X via Z, node X calculates the throughput pheromone value $\delta^x(w,z)$ *to* node W through node Z using:

$$\delta^x(w,z) = C + \delta^x(w,z) \tag{13}$$

where C is a constant. Node X maintains a routing table, as shown in Table II, containing a throughput-pheromone δ levels. Pheromone values will decrease in time in the absence of positive feedback. The periodic decay for the throughput-pheromone is governed by the iteration:

$$\delta^x(w,z) = \beta\delta^x(w,z) \tag{14}$$

where $0 < \beta < 1$. For a highly volatile network, we can increase the frequency of decay or decrease the β value. This ensures that stale routes decay faster.

The data packets are routed according to the probabilities that are simply proportional to throughput-pheromone levels. Specifically, data packets at X destined to node W are routed through Z with probability:

$$p^x_\delta(w,z) = \frac{\delta^x(w,z)}{\delta^x(w,z)\ + \delta^x(w,y)\ + \delta^x(w,u)}. \tag{15}$$

If data packets are dropped partially (e.g., Jellyfish Attacks [2]) by a malicious node, the destination will receive fewer data packets and therefore send BANT packets back to the source less frequently along the path that contains the malicious node. Therefore, at the source, the corresponding pheromone level will slowly increase compared to the pheromone levels of other paths. Similarly, if data packets are dropped completely (e.g., Blackhole Attacks [2]) by a malicious node. The destination has no cause to send BANT packets to the source along that path. Therefore, the corresponding pheromone level will stop increasing, and the path will be avoided. So, under ant-based multipath routing, malicious packet dropping attacks (either Jellyfish or Blackhole attacks) are naturally avoided. Note that our scheme can avoid any attack that degrades the throughput of a path. Conventional single path routing algorithm cannot achieve this property. Even if an application layer sends back an acknowledgement to the source, the source cannot avoid the malicious node at the rerouting phase. Further more, the acknowledgement could be maliciously dropped.

Two attack scenarios were considered in our simulation study. First, we simulated Jellyfish [14] attacks. Specifically, we assumed, among three paths, one contained one malicious node and another contained two malicious nodes. The malicious nodes were assumed to have benignly cooperated in the routing phase to build multiple routes between source and destination. However, subsequent data packets and BANT packets were dropped discreetly without an error message sent back to the source. Therefore, the source had no knowledge of malicious packet drops. Generally, however, a source receives an error message when normal packet loss occurs[3], and it considers the path to be broken.

We simulated two different dropping rates ($\rho = 0.05$ and $\rho = 0.10$). Discreet intruders with small dropping rates were selected in order to validate our scheme. A malicious node with large dropping rate can easily be detected and avoided. Secondly, we simulated Jellyfish and Blackhole [14] attacks together. A Blackhole attack is a special case ($\rho = 1.00$) of a Jellyfish attack, i.e., all packets are simply dropped. Specifically, we assumed that one path contained a malicious node performing Jellyfish attacks and another contained a malicious node performing Blackhole attacks. Unlike the first scenario, all nodes were assumed to be secure initially. The node that performed Jellyfish attacks was compromised at 100 seconds, and the node that performed Blackhole attacks was compromised at 200 seconds.

Path throughput, under the first scenario, is depicted in Figure 11, where throughput was calculated every 10 seconds. Path 1 contained one malicious node, path 2 was secure and path 3 contained two malicious nodes. The average throughput of each path is represented by straight line with corresponding line type. For both Figure 11(a) and 11(b), we see path 2 became dominant. Note the convergence speed was observed to be roughly proportional to the dropping rate ρ. In particular, the throughput of path 3 decreased faster than that of path 1 as expected. Since path 3 contained more malicious nodes, it dropped more data and BANT packets resulting in a lower rate of pheromone table update. Other performance metrics, under Jellyfish attacks, are summarized in Table III.

TABLE III
CASE 1 : JELLYFISH ATTACKS

Drop Rate	Packet Delivery Ratio (%)	End-to-End Delay (msec)	Routing Overhead (%)
0.00	99.98	32.6	10.04
0.05	96.83	32.8	10.02
0.10	99.40	36.8	10.01

Similarly, path throughput, under the second scenario, is depicted in Figure 12. The secure path in this experiment was path 1. One of the nodes of path 2 became malicious at 100 seconds and performed Jellyfish attacks. Again, two different dropping rates ($\rho_1 = 0.05$ and $\rho_1 = 0.10$) were considered. At 200 seconds, one of the nodes of path 3 started to drop all data/BANT packets (Blackhole attack, $\rho_2 = 1.00$). We see the throughput of path 3 dropped to zero at 200 seconds. As expected, for both Figure 12(a) and 12(b), path 1 became dominant. Other performance metrics, under Jellyfish and Blackhole attacks, are summarized in Table IV.

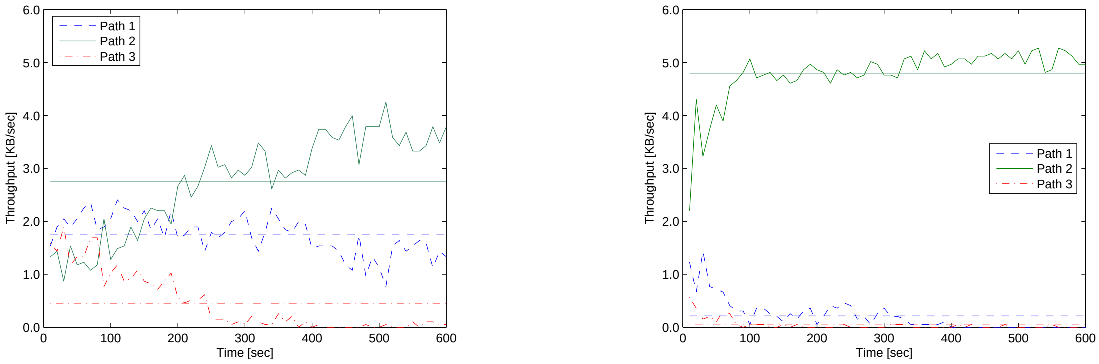
TABLE IV
CASE 2 : JELLYFISH AND BLACKHOLE ATTACKS

Drop Rate	Packet Delivery Ratio (%)	End-to-End Delay (msec)	Routing Overhead (%)
0.05	97.02	28.6	10.05
0.10	96.55	27.8	10.03

VII. CONCLUSIONS

In this paper, we proposed a multipath QoS routing protocol that can accommodate *both* latency-critical and latency-non-critical traffic under volatile network conditions. The mechanism was based on information obtained from periodically transmitted ants resulting in reinforced path-pheromone levels. For latency-critical traffic, routing decisions were based on combinations of *normalized* energy and delay pheromone levels (9). For latency-non-critical traffic, only energy pheromone levels were used. With the help of PPRA, latency-non-critical traffic was transmitted over the most energy rich path, while latency-critical traffic was transmitted over low delay paths that are also energy rich for reliability (9), and the network lifetime was maximized. In simulation studies our algorithm rapidly responded to network volatility maintaining good delay and throughput performance with a comparable amount of (ant) overhead, that itself required little in the way of computation and no specialized hardware. In particular, for additional overhead, the algorithm showed good minimum energy performance which is an important factor of maximizing network lifetime. Also, the algorithm was applied to a security context in order to avoid malicious packet dropping. The mechanism was based on throughput information obtained from periodically transmitted BANTs resulting in reinforced path-pheromone levels.

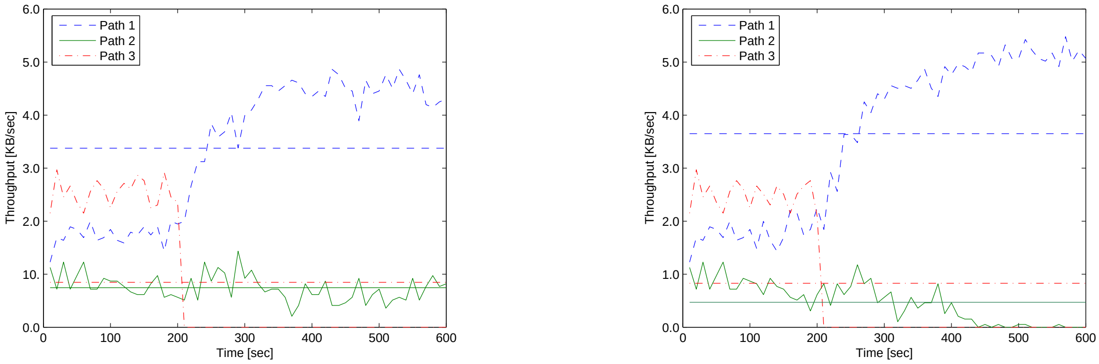
[3] Normal packet loss is caused by various reasons such as congestion (buffer overflow), excessive collision (limited retransmission), topology change, etc.



(a) $\rho = 0.05$

(b) $\rho = 0.10$

Fig. 11. Path Throughput under Jellyfish Attacks



(a) $\rho_1 = 0.05, \rho_2 = 1.00$

(b) $\rho_1 = 0.10, \rho_2 = 1.00$

Fig. 12. Path Throughput under Jellyfish and Blackhole Attacks

REFERENCES

[1] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[2] I. Aad, J. Hubaux, and E. Knightly. Denial of service resilience in ad hoc networks. In *Proceedings of ACM MOBICOM*, Philadelphia, PA, Sep. 2004.

[3] Mohammed S. Al-kahtani and Hussein T. Mouftah. Enhancements for clustering stability in mobile ad hoc networks. In *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 112–121, Montreal, Quebec, Canada, Oct. 2005.

[4] P. Arabshahi, A. Gray, I. Kassabalidis, A. Das, S. Narayanan, M. El-Sharkawi, and R. Marks II. Adaptive routing in wireless communication networks using swarm intelligence. In *Proceedings of the 9th AIAA Int. Communications Satellite Systems Conf.*, pages 17–20, Toulouse, France, Apr. 2001.

[5] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, Sep. 2002.

[6] C. Balasubramanian and J.J. Garcia-Luna-Aceves. Shortest multipath routing using labeled distances. In *Proceedings of IEEE MASS*, Oct. 2004.

[7] S. Bouam and J. Othman. Data security in ad hoc networks using multipath routing. In *Proceedings of 14th IEEE PIMRC*, Beijing, China, Sep. 2003.

[8] M. Burmester and T. Le. Secure multipath communication in mobile ad hoc networks. In *Proceedings of IEEE ITCC*, Las Vegas, NV, Apr. 2004.

[9] J. Cano and D. Kim. Investigating performance of power-aware routing protocols for mobile ad hoc networks. In *Proc. of the International Mobility and Wireless Access Workshop*, pages 80–86, Forworth, Texas, USA, Oct 2002.

[10] G. D. Caro and M. Dorigo. Mobile agents for adaptive routing. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences(HICSS)*, Kohala Coast, HI, USA, 1998.

[11] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications, Special Issue on Self-organization in Mobile Networking*, 16:443–455, 2005.

[12] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM*, Tel-Aviv, Israel, Mar 2000.

[13] S. Chen and K. Nahrstedt. On finding multi-constrained paths. In *Proceedings of IEEE ICC'98*, pages 874–879, Atlanta, GA, USA, Jun 1998.

[14] H. Deng, W. Li, and D. Agrawal. Routing security in ad hoc networks. *IEEE Communications Magazine*, 40:70–75, Oct. 2002.

[15] S. Doshi, S. Bhandare, and T. Brown. An on-demand minimum energy routing protocol for a wireless ad hoc network. *MC2R*, 6(3), 2002.

[16] G. Feng, K. Makki, N. Pissinou, and C. Doulgeris. An efficient approximate algorithm for delay-cost-constrained QoS routing. In *Proceedings of ICCCN*, pages 395–400, Phoenix, AZ, USA, Oct 2001.

[17] K. Fujita, A. Saito, T. Matsui, and H. Matsuo. An adaptive ant-based routing algorithm used routing history in dynamic networks. *IPSJ SIGNotes MoBiLe computing and wireless communications*, 2001.

[18] M. Gunes, U. Sorges, and I. Bouazizi. ARA - the ant-colony based routing algorithm for MANETs. In *Proceedings of ICPPW*, Vancouver, B.C., Canada, Aug. 2002.

[19] Y. Hu, D. Johnson, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the IEEE WMCSA*, 2002.

[20] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of ACM MOBICOM*, Atlanta, GA, Sep. 2002.

[21] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE INFOCOM*, 2003.

[22] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of ACM WiSe*, 2003.

[23] O.H. Hussein, T.N. Saadawi, and Myung Jong Lee. Probability routing algorithm for mobile ad hoc networks' resources management. *IEEE JSAC*, 23:2248–2259, Dec. 2005.

[24] P. Jeon and G. Kesidis. Avoiding malicious packet dropping in ad hoc wireless networks using multipath routing. In *Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, Urbana, IL, USA, Sep. 2005.

[25] P. Jeon and G. Kesidis. Pheromone aided robust multipath and multi-priority routing in wireless MANETs. In *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 106–113, Montreal, Quebec, Canada, Oct. 2005.

[26] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[27] M. Just, E. Kranakis, and T. Wan. Resisting malicious packet dropping in wireless ad hoc networks. In *Proceedings of ADHOC-NOW*, Montreal, Canada, Oct. 2003.

[28] Y. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[29] T. Korkmaz and M. Krunz. Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks. *Modeling and Computer Simulation*, 10(4):295–325, 2000.

[30] P. Kotzanikolaou, R. Mavropodi, and C. Douligeris. Secure multipath routing for mobile ad hoc networks. In *Proceedings of WONS*, 2005.

[31] R. J. La, J. H. Yoo, and M. Makowski. Convergence results for ant routing. In *Proceedings of CISS*, Mar. 2004.

[32] S. J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *ICC 2001*, volume 10, pages 3201–3205, Jun 2001.

[33] Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata. Biologically inspired self-adaptive multi-path routing in overlay networks. *Communications of the ACM*, 49(3):62–67, Mar. 2006.

[34] X. Li and L. Cuthbert. Stable node-disjoint multipath routing with low overhead in mobile ad hoc networks. In *Proceedings of IEEE MASCOTS*, Oct. 2004.

[35] M. Marina and S. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of the IEEE ICNP*, Riverside, CA, Nov. 2001.

[36] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, Nov 2001.

[37] S. Medidi, M. Medidi, S. Gavini, and R. Griswold. Detecting packet mishandling in manets. In *Proceedings of SAM*, Las Vegas, NV, Jun. 2004.

[38] R. Muraleedharan and L. Osadciw. Balancing the performance of a sensor network using an ant system. In *37th Annual Conference on Information Sciences and Systems*, Baltimore, MD, USA, Mar 2003.

[39] C. Murthy and B. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, May 2004.

[40] A. Nasipuri and S. R. Das. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of the 8th Int. Conf. On Computer Communications and Networks (IC3N)*, Moston, MA, October 1999.

[41] P. Papadimitratos and Z. Haas. Secure data transmission in mobile ad hoc networks. In *Proceedings of WiSe*, San Diego, CA, Sep. 2003.

[42] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM*, Kobe, Japan, April 1997.

[43] C. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, Louisiana, USA, Feb. 1999.

[44] R. Rao and G. Kesidis. Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited. In *Proceedings of Globecom*, Dec. 2003.

[45] M. Roth and S. Wicker. Termite: Emergent ad-hoc networking. In *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks*, Mahdia, Tunisia, June 2003.

[46] A. Safwat, H. Hassanein, and H. Mouftah. Energy-aware routing in manets: Analysis and enhancements. In *Proceedings of MSWiM*, Atlanta, GA, USA, Sep 2002.

[47] G. Sager. Security fun with OCxmon and cflowd. In *Internet 2 Working Group Meeting*, Nov. 1998.

[48] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Royer. A secure routing protocol for ad hoc networks. In *Proceedings of IEEE ICNP*, Paris, France, Nov. 2002.

[49] Chien-Chung Shen and Chaiporn Jaikaeo. Ad hoc multicast routing algorithm with swarm intelligence. *Mobile Networks and Applications*, 10(1-2):47–59, Feb. 2005.

[50] S. Singh and C. Raghavendra. Pamas: Power aware multi-access protocol with signaling for ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 26(3):5–25, Jul 1998.

[51] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer. Hash-based IP traceback. In *Proc. ACM SIGCOMM*, 2001.

[52] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer. Single-packet ip traceback. *IEEE Trans. Networking*, Dec. 2002.

[53] D. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proceedings of IEEE INFOCOM*, 2001.

[54] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of MOBICOM*, San Diego, CA, USA, Sep 2003.

[55] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *Proceedings of SIGCOMM*, Aug. 2002.

[56] D. Subramanian, P.Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proceedings of IJCAI*, Aug. 1997.

[57] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8:153–167, 2002.

[58] Horst F. Wedde, Muddassar Farooq, Thorsten Pannenbaecker, Bjoern Vogel, Christian Mueller, Johannes Meth, and Rene Jeruschkat. Beeadhoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 153–160, Washington DC, USA, June 2005.

[59] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205, New York, NY, USA, June 2002.

[60] K. Wu and J. Harms. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of European Personal and Mobile Communications Conference (EPMCC)*, Vienna, Austria, Feb. 2001.

[61] K. Wu and J. Harms. Performance study of a multipath routing method for wireless mobile ad hoc networks. In *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'01)*, pages 99–107, Washington, DC, USA, Aug. 2001.

[62] K. Wu and J. Harms. Multipath routing for mobile ad hoc networks. *Journal of Communications and Networks Special Issue on Innovations in Ad Hoc Mobile Pervasive Network*, 4(1):48–58, March 2002.

[63] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, New York, NY, USA, Jun 2002.

[64] Z. Ye, S. Krishnamurthy, and S. Tripathi. A framework for reliable routing in mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, Mar. 2003.

[65] Yuan yuan Zeng and Yan xiang He. Ant routing algorithm for mobile ad-hoc networks based on adaptive improvement. In *Proceedings of IEEE WCNM*, volume 2, pages 678–681, Wuhan, China, Sept. 2005.

[66] Saida Ziane and Abdelhamid Melouk. A swarm intelligent multi-path routing for multimedia traffic over mobile ad hoc networks. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 55–62, Montreal, Quebec, Canada, Oct. 2005.

Paul Barom Jeon Paul Barom Jeon received his M.S. in Electronic Engineering from Sogang University, Korea in 1996. He was a research engineer in the Mechatronics Center of the Samsung Heavy Industries, Korea, from 1996 to 2001. He received his Ph.D. in Electrical Engineering from Pennsylvania State University in 2006. His research areas span QoS multipath routing, network protocol design, and security. Currently, he is a student member of the IEEE.

George Kesidis George Kesidis received his M.S. and Ph.D. in EECS from U.C. Berkeley in 1990 and 1992 respectively. He was a professor in the E&CE Dept of the University of Waterloo, Canada, from 1992 to 2000. Since April 2000, he has taught in both the CS&E and EE Departments of the Pennsylvania State University. His research experience spans several areas of computer/communication networking including security, incentive engineering, efficient simulation, and traffic engineering. Currently, he is a senior member of the IEEE and the TPC co-chair of IEEE INFOCOM 2007.