A Cross-layer Design for Bee-Inspired Routing Protocols in MANETs

Giagkos, Alexandros; Wilson, Myra S.

Publication date:
2009

Citation for published version (APA):
Giagkos, A., & Wilson, M. S. (2009). *A Cross-layer Design for Bee-Inspired Routing Protocols in MANETs*. 25-32. http://hdl.handle.net/2160/3027

A Cross-layer Design for Bee-Inspired Routing Protocols in MANETs

Alexandros Giagkos and Myra S. Wilson
aag07@aber.ac.uk, mxw@aber.ac.uk

***Abstract*— The field of robotics relies heavily on various technologies such as mechanical and electronic engineering, computing systems, and wireless communication. The latter plays a significant role in the area of mobile robotics by supporting remote interactions. An effective, fast, and reliable communication among homogeneous or heterogeneous robots, as well as the ability to adapt to the rapidly changing environmental conditions predicates the robots success and completion of their tasks. In this paper we present our research position in the area of adaptive nature-inspired routing protocols for mobile ad hoc networks (MANETs). Our approach is based on the honeybee foraging behaviour and ability to find and exchange information about productive sources of food in a rapidly changing environment. We describe the research problem, present a brief review of the relative literature, and illustrate our future plan.**

***Index Terms*— wireless, mobile, ad hoc, bee-inspired, cross-layering, routing.**

I. INTRODUCTION

In wireless communication, apart from the classical hidden and exposed terminal problems, nodes in an ad hoc environment face two other major challenges: the mobility of the network participants and the resource constraints they have.

Due to the mobility of nodes, the network topology is highly dynamic and all traffic suffers from frequent path breaks. The routing protocols of such networks must be able to perform efficiently and effectively, adapting to any topological changes. In robotics, hardware tends to be lightweight, portable and, depending on the project, occasionally cheap. Thus, the protocols must also optimally manage any local (battery capacity, processing power, etc.) and globally (bandwidth of links, transmission ranges, etc.) available resources.

A plethora of protocols have been proposed in order to satisfy the above challenges, some of which are presented in the background section of this paper. However, as the number of mobile robotic applications increases, the problems described above are also increased. This results in a demand for new intelligent approaches, not necessarily based on traditional disciplines. In this paper we propose a solution that is inspired by the study of honeybees. The collaborative behaviour of bees not only achieves a remarkable channel of simple communication between them, but also provides a medium for the exchange of complex knowledge for the sake of their colony.

The rest of this paper is structured as follows: section II is a literature review and section III presents our research objectives and hypotheses. In section IV we give a view of what the study of real honeybees has given to networks by looking at an existing approach. Section V is an introduction to cross-layering and an illustration of how it can be used in our research. Section VI outlines our next steps, while sections VII and VIII refer to the network simulator and evaluation framework we plan to use, respectively. This paper concludes with section IX.

II. BACKGROUND

The major contribution of this section is to provide a literature review of existing adaptive, state-of-the-art algorithms that have been proposed and used in mobile ad hoc networks. Table 1 illustrates the classification of the algorithms discussed by this paper.

A. Adaptive routing algorithms

Mobile ad hoc routing protocols can be generally categorized as proactive (table-driven protocols), reactive (on-demand protocols), and hybrid (proactive and reactive protocols).

The main characteristic of a proactive protocol is that all nodes within a network maintain at least one routing table which describes the whole or, in some cases, only a part of the topology. Whenever a new path to a destination is required, nodes run a shortest path finding algorithm using their routing table in order to find an appropriate route.

One of the first proposed routing protocols for ad hoc wireless networks was the destination sequenced distance-vector routing protocol (DSDV) [1]. This table-driven protocol is an enhanced version of the distributed Bellman-Ford algorithm and implies that all destinations are readily available at every node of the network, at all times. This is achieved by exchanging routing tables, whose contents include the shortest distance and the first node on the shortest path to every other node of the network. These tables are exchanged and updated at regular intervals and also when a node observes a significant change in the local topology.

Once a routing table has been passed to a neighbouring node, an incremental update or full dump may occur. The first is used when a node does not observe significant changes in the local topology, whereas the second is done either when the local topology changes dramatically or when an incremental update requires more than a single network protocol data unit (NPDU). When a node receives new routing information, that information is compared to the information already available from previous routing packets. The comparison is made using the idea of marking each

Alexandros Giagkos and Dr Myra S. Wilson are with the Intelligent Robotics Group, Dept. of Computer Science, Aberystwyth University, Penglais, Aberystwyth, Ceredigion, Wales, UK, SY23 3DB.

shortest path with a sequence number. Any incoming route with a more recent sequence number is immediately adopted, while routes with older sequence numbers are discarded from the table. In the case where two routes have equal sequence numbers, the one with the best metric score is selected. The best metric may be the lowest number of hops.

The advantage of DSDV is its ability to provide routes to all destinations at all times to any full dump update, ensuring less delay compared to other methods during node's set-up process. However such behaviour increases the protocol's overhead, especially under conditions with high mobility.

The optimized link state routing (OLSR) [2] is another proactive routing protocol. It is an optimization of the pure link state algorithm adapted to the requirements of the mobile network. The key concept used in this method is that of multipoint relays (MPRs). A MPR is a selected node which forwards broadcast messages during the flooding process of each routing. Using this technique, OLSR reduces the size of the control packets as well as minimizes the control traffic flooding. OLSR consists of three main components: the HELLO messages, the topology control (TC) messages, and the multipoint relay selection and signalling.

Starting from the first one, a HELLO message is periodically sent by nodes and includes its own address accompanied by three lists of neighbouring nodes (adjacent nodes, nodes that have been part of a successful bidirectional connection, and neighbours that have been selected to act as MPRs). Hence, using HELLO messages the nodes in the mobile network are able to exchange neighbouring nodes' addresses and find possible positive pairs of connection. Like HELLO messages, a TC message is used to periodically exchange information about the current topology. The payload of a TC message is a set of bidirectional links between a node and its neighbours. This information is very important for the MPR selection and the signalling phase of the protocol explained below.

All nodes must select a MPR within their neighbourhood. The selection criterion of a MPR is that a message sent by any node should be repeated by the MPR and received by all nodes in the neighbourhood two (2) hops away. The information required for the selection as well as the MPR advertisements are collected and propagated by the HELLO messages.

The advantage of OLSR over other proactive protocols such as DSDV, is that it not only reduces the routing overhead associated with table-driven routing, but it also reduces the number of broadcasts done. On the other hand, OLSR remains prone to the common proactive routing vulnerabilities such as incorrect control traffic generation and relaying [3].

Unlike proactive protocols, on-demand protocols do not maintain routes between all the nodes in an ad hoc network. Rather, routes are established when needed through a route discovery process in which a route request is broadcast. A route reply is returned either by the destination or by an intermediate node with an available route.

One of the most widely used and well-known examples is the adaptive on-demand distance vector (AODV) [4] protocol. Although it is based on a distance vector routing, AODV is designed to request for a route only when the route is needed. Additionally, AODV does not require nodes to maintain routes to destinations that are not actively used, reducing in that way both control messages overhead and power consumption.

The way this protocol provides routing solutions relies on the following concept. A node broadcasts a route request (RREQ) message when it determines that it needs a route to a destination and does not already have one available. Such a situation can occur when the destination node is unknown or a previously valid route has expired. Hence expiry time of each route in the routing table of a node plays a significant role to the routing mechanism. When a node receives a RREQ message it may send a route reply (RREP) back, if it is either the destination of it or it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. Otherwise, it rebroadcasts the RREQ to others. Every time a route is used to forward a packet the expiry time of that route gets updated. In case of a broken link, nodes are notified with a route error (RERR) message, which gets propagated to all nodes in order to invalidate the broken route.

Clearly, the main advantage of this protocol is that routes are established on demand, which offers less delay during connections set-up. Contrarily though, the protocol's dependence on sequence numbers can drive the system instability. Such a situation may happen when an out-of-date intermediate node has a higher but not latest destination sequence number for a path.

Dynamic source routing (DSR) [5] is another example of reactive routing protocols. A routing entry in DSR contains all intermediate nodes to be visited by a packet rather than just the next hop information maintained by DSDV or AODV. In other words, if the source knows the exact routing path to the destination, it puts it in the data packet. In a different case, it performs a routing discovery by flooding the network with a RREQ message. The RREQ message gets updated as it visits the intermediate nodes on its way to the destination. If it has an answer, a node can always reply with a RREP message, by completing the route it receives from the latest RREQ.

The overhead of the routing discovery is reduced by DSR, as each node monitors and uses the RREQs to update their local caches (promiscuous mode). One of the disadvantages of this protocol is that the route maintenance mechanism does not locally repair a broken link, and stale route cache information could also result in inconsistencies during the route discovery.

Another beacon-based on-demand algorithm is used by the associativity-based routing ABR [6] protocol. In ABR, a source node floods RREQ packets throughout the network if a route is not available in its local route cache. The RREQ packets are forwarded by all the intermediate nodes, carrying the path they have traversed and the sequence number of the beacon. When the first RREQ reaches the destination, the destination waits for a time period to receive multiple RREQs through different paths. The reason why ABR is an associativity-based algorithm is that after a duration, the destination selects the path that has the maximum proportion of stable links. Special rules are applied in case two paths have the same proportion of stable links: the shorter is selected. In the case of having more than one shortest path, a random path is selected as the path between the source and the destination.

One major advantage of this protocol is that stable routes have a higher preference compared to shorter routes. This results in less path breaks which reduce the extent of flooding due to the reconfiguration of paths. An immediate disadvantage is that the chosen path may be longer than the shortest path.

Hybrid protocols in wireless ad hoc networks are any adaptive routing protocols that combine the best features from both reactive and proactive techniques. Generally, hybrid protocols separate the network topology in zones. Routing is determined proactively within each zone, and reactively outside it. A general advantage of such a combination is the increased overall scalability and optimization within the zones.

One well-known hybrid example is the zone routing protocol (ZRP) [7]. As in most of the hybrid protocols, the key concept employed in ZRP is to use a proactive routing method within a limited zone (of a predefined number of hops neighbourhood of nodes), and a reactive one for nodes beyond this zone. Each node maintains the information about routes to all nodes that belong to the same zone by exchanging periodic route update packets. On the other hand, the reactive method is responsible for finding paths to the nodes which are not within the routing zone of a node. If a packets destination is outside the zone, the reactive method broadcasts a RREQ message to its peripheral nodes. RREP messages are also used to answer to the RREQs.

By combining the best features of proactive and reactive routing methods, ZRP reduces the control overhead compared to the RREQ flooding mechanism of the pure on-demand, and the periodic flooding of the pure table-driven protocols. Also, the decision on the zone radius has a significant impact on the performance of the protocol.

B. Nature-inspired algorithms

This section reviews a number of different nature-inspired protocols, that is, protocols that provide routing facilities by applying concepts borrowed from Nature. These concepts are usually derived by the broad field of Swarm Intelligence [8].

AntHocNet is a hybrid multipath algorithm. It explores the capabilities of the Ant Colony Optimization (ACO) [9], combining them with both reactive and proactive ways of gathering and building routing data. In more detail, when a data session is started between a source and a destination node, the source checks whether it has up-to-date routing information for the destination. If it has not, it reactively sends out an ant-like agent, called reactive forward ant, in order to look for paths to the destination. Therefore a forward ant is used to gather information about the quality of the path it follows. Once it reaches the destination, it traces back the path to the source node, updating the routing tables in its path. On its way back, the forward ant becomes backward ant.

In AntHocNet [10] a routing table consists of a destination, the next possible hop to it, and a pheromone. A pheromone is a value that indicates the estimated goodness of a path between a source and a destination. In this way, pheromone tables in different nodes indicate multiple paths between two nodes in the network, and are stochastically spread over it (in each node they select the next hop with a probability proportional to its pheromone value). Once paths are set up and the data start to flow, the source node starts to send proactive forward ants to the destination. This is a maintenance phase where each proactive forward ant follows the pheromone values in the same ways as the data, but has a small probability at each node of being broadcast. This technique serves as follows. If the forward ant reaches the destination without a single broadcast, it means that the current path is working, and it provides an efficient way of data transferring. On the other hand, if the ant gets broadcast at any point, it leaves the currently known pheromone trails and it explores new paths. A threshold of value two (2) is used to avoid proactive forward ants being broadcast to the whole network, allowing the search for improvements or variations to be concentrated around the current paths. In the case of a link failure, a node may use an alternative path based on the pheromone values. However, if the failed link was the only one in each pheromone table, the node sends out a route repair ant that travels to the involved destination like a reactive forward ant would do. Simulation experiments have shown that AntHocNet can outperform AODV in terms of delivery ratio and average delay [10].

Another biologically inspired algorithm that tries to address the problem of routing is used in the Termite protocol [11]. Termite takes into consideration the ability of social insects to self organize, and is based on the concept of the termite hill building [12, explained in detail]. Hill building illustrates the four principles of self-organization of the ant-like societies, i.e., positive feedback, negative feedback, randomness, and multiple interactions.

Termite associates a specific pheromone scent with each node in the network. Packets moving through the network are biased to move in the direction of the pheromone gradient of the destination node, exactly as biological termites are biased to move towards their hill. Positive feedback is gained from links that have stronger pheromone scent, whereas negative feedback is represented by pheromones evaporation. The randomness factor is used for termites exploring the network for the first time, and of course multiple interactions are achieved by having multiple termite agents exchanging information as they pass through intermediate nodes.

The argument of Termite is that the Swarm Intelligence framework can be used to competitively solve the routing problem in mobile wireless ad hoc networks, with a minimal use of control overhead. A small amount of control information is piggybacked in every data packet, which is usually sufficient for the network to maintain a current and accurate view of its state. Using that information, the routing algorithm is able to generate routing decisions.

Bees and the wisdom of the hive have also been a source of inspiration to researchers. BeeSensor [13] is a bee-inspired power aware routing protocol which utilizes a simple bee-agent model and requires little processing and network resources. It is based on the idea of using three types of bee-agents packers, scouts, and foragers. A packer is the component of the BeeSensor protocol responsible for receiving or sending back data between the application layer and an appropriate forager (see below). This bee-agent is not actually transmitted, rather it works internally within a node. A scout on the other hand, is being transmitted and can be classified as forward scout or backward scout, depending on the direction by which it travels. Similarly to the ant-like protocols, a source that requires a route to a destination that does not already posses, broadcasts a forward scout. Each forward scout carries its unique ID and the initial event in the payload (i.e., the reason for it being sent). Once a forward scout reaches its destination, it changes to a backward scout and returns to its source. Once it is back, the wisdom of the hive is applied. A dance number is calculated by taking into consideration the energy of the path gathered by the scout. The result of the dance number indicates the number of foragers to be cloned from this scout. A forager is the main worker of the BeeSensor protocol, that is, the bee-agent which transports data from the source node to the destination. As already described, a forager receives data from a packer within the protocol (hive) and travels over the network in order to reach the destination in a hop-by-hop manner.

In [13] the authors show that BeeSensor delivers better performance in a dynamic wireless sensor network (WSN) [14] or in MANETs as compared with the optimized versions of AODV, while its computational and bandwidth requirements are significantly smaller.

TABLE I
CLASSIFICATION OF ROUTING ALGORITHMS

Name	Type
ABR	Reactive
AntHocNet	Nature-inspired / Hybrid
AODV	Reactive
BeeHive	Nature-inspired / Hybrid
BeeSensor	Nature-inspired / Hybrid
DSDV	Proactive
DSR	Reactive
OLSR	Proactive
Termite	Nature-inspired / Hybrid
ZRP	Hybrid

III. RESEARCH OBJECTIVES AND HYPOTHESES

Our research problem can be outlined as follows: *we need to develop a decentralized, dynamic, reliable, and robust bee-inspired routing protocol in a cross-layered fashion, to provide cheap routing solutions for a wireless ad hoc network of mobile nodes.*

A. Objectives

What we believe that is to be achieved by our approach can be summarized as:

- Directly map concepts of biology and honeybee behaviours to the technical system of network routing.
- Use information taken from lower layers other than the network layer to enhance the evaluation of the communication links.
- Use the collaborative behaviour of simple agents to build enough knowledge in order to provide effective routing results.
- Enhance the current techniques of applying the wisdom of the hive to the network communication field, by extending their capability to mimic Nature.

B. Research hypotheses

The literature we are aware of at the moment shows that applying the honeybees' remarkable sophistication and communication capabilities to the domain of networking, has achieved quite competitive status between other strategies of tackling the routing problem.

A honeybee colony has a number of features that are desired in networks. Communication is decentralized, information is being transferred from one member of the colony to others only when they are back in the hive, the exchanged information contains details about the distance and the quality of the food source, etc. [15]. Keeping in mind what existing bee-inspired approaches have proved, in our work we will use the following hypotheses:

H1: If the method used by a honeybee colony in order to evaluate the goodness of a food source is a corporate consideration of a number of factors, which indicate different attributes of the food source, then an artificial colony based on similar principles should be able to obtain analogous information, and use it to evaluate the goodness of any neighbouring node as part of a routing solution. Providing routing solutions should be flexible, scalable, reliable and efficient because its natural counterpart has all these features.

H2: If the profitability of the food source determines the number and liveliness of the dances, thus the number of foragers recruited in a honeybee colony, then monitoring the level of profitability of each routing path in the artificial system could also be used to affect the probability of a path being selected as the next forwarding hop.

IV. FROM NATURE TO NETWORK

In this section, we present the concepts of honeybee foraging and decision making for bee dancing, from a biological point of view. We also examine how that knowledge can be mapped to the network domain and exploited further, in order to achieve competitive network routing results.

A. Natural honeybee behaviour

In Nature, a honeybee explores the surroundings of the hive in order to detect possible sources of food [15]. Once a source is found, the scout (as honeybees are called in this case) returns back to the hive to report her findings, and to recruit other hive members to start foraging. Both reporting and recruiting are done by performing a special dance. This dance involves touching, tasting fluids and pollen from the performer's stomach and body respectively, alterations to the dancing tempo (by altering the frequency of moving the wings), and special round movements occupying and crossing cells of the dance floor[1] [16].

Biologists have worked for more than fifty years exploring the dance of bees as a medium of communication. They have shown that there are two ways of bee dancing: the round dance, which is performed when the source of food is a short distance from the hive, and the tail-wagging dance, which is for long distances. Both of these happen over naturally fixed intervals, and are used to pass useful pieces of information to the prospective forager bees, such as the kind and the quality of the source, its distance from the hive, and an accurate enough direction (angle) to it.

Nevertheless, a honeybee does not just start dancing every time she returns back to the dance floor. In [17], Seeley defines the bee's criterion of profitability in an attempt to answer the question of *how does a bee measure the profitability of a food source?* Seeley introduces the idea of energetic profitability and after a series of experiments, he came to the conclusion that honeybees judge according to it. If during an average foraging trip, a honeybee collects G units of energy, expends C units of energy, and spends time T, then to maximize the net rate of energy collection she should maximize:

$$\frac{(G-C)}{T}$$

and to maximize net energetic efficiency she should maximize:

$$\frac{(G-C)}{C}$$

Besides the facts that Seeley has empirically proved, in his book [16] von Frisch presented the understanding of the dependence of the honeybee dances on the profitability of foraging activity. He showed that although the pattern of the honeybee's dance is determined fundamentally by the distance of and direction to a source of food, whether the bee's dancing will take place depends on many factors that significantly regulate the relation between supply and demand. These factors affect not only the honeybees' decision, but also the number of foragers being recruited. The most promising of which, i.e., have the potential of being used in our research work, are the following:

a) The sweetness of the sugar solution. A sugar solution must have a certain concentration if it is to be ingested by the honeybee. Depending on conditions, there is an acceptance threshold that is used in order to determine whether a source of food is accepted or not.

b) The purity of the sweet taste. A pure sucrose solution, near the acceptance threshold in strength, releases more dances than one slightly contaminated with other non-sweet ingredients.

c) Ease of obtaining the solution. The required expenditure if time plays a significant role in the honeybee's decision to perform her dance.

d) Nearness of the food source. Given the same quality of food, a nearby feeding place should release more dancing than a more distant one, resulting in a bigger number of foragers during the recruitment.

e) Improvement of the food. Aside from the fact that better sugar solution releases more dancing, a proof of improvement of a path as such has a dance-promoting effect.

Furthermore, the profitability of a food source has another interesting effect in honeybee colonies. Having in mind that the reason for dancing is the recruitment of foragers, von Frisch has shown that the goodness or badness of a food source also determines the liveliness of the dance. Thus, the number of honeybees being alerted. In more detail, when the food supply is plentiful nearly all foragers dance vigorously and their dance lasts for longer time. On the other hand, if the quantity of the sugar content of the food diminishes, the dances grow less and become shorter, making the influx of new honeybees decrease. If matters become yet worse, and the quality of food becomes unacceptable (or has been naturally moved), the original foragers terminate their activity.

B. Concepts in BeeHive

The BeeHive [18], introduced by M. Farooq, is a routing algorithm inspired by the communicative and evaluative methods and procedures of honeybees. The design of BeeHive enables intelligent bee-agents to explore network regions, or *foraging zones*, collecting information which is then deposited to local routing tables. The reason why BeeHive is presented in a separated section of this paper, is because it can be considered as the fundamental work in the area of bee-inspired network engineering. Additionally, BeeHive has been extensively tested and evaluated. Its results conclude that while it achieves similar or better performance compared to state-of-the-art routing algorithms, honeybee agents occupy smaller bandwidth and require significantly less process time compared to the agents of existing algorithms. BeeHive has been an inspiration to further research and enhancements.

Initial part of this protocol's design is the direct mapping of some biological concepts to networks.

- Each node in the network is considered as a hive that consists of honeybee agents that can behave exactly as real honeybee scouts do.
- These artificial scouts provide the nodes they visit with information about the propagation delay and queuing delay of the paths they explored. These two pieces of knowledge are then used internally to estimate the goodness of a neighbouring node.
- A honeybee agent decides whether to provide its path information to a node based on the quality of the path, examined against a threshold. This threshold depends

---

[1] This place exists in every hive and is where honey bees either wait or dance.

on the number of hops a honeybee agent is allowed to take.

- A routing table is considered as the dance floor where honeybee agents provide information about the quality of the paths.
- The quality of paths is mapped onto the quality of nodes. Consequently, the quality of a node is formulated as a function of proportional quality of only those neighbouring nodes that possibly lie in the path towards the destination.
- Data packets in the network are considered as foragers. They also access the information in the routing tables of any node they visit, which enables them to become aware of the quality of different neighbouring nodes for reaching their destinations. Thus, they select the next neighbouring node toward the destination in a stochastic manner, depending upon its goodness.

From the above mapping rules we can observe a couple of interesting assumptions made by BeeHive. To begin with, there are two criteria present for deciding whether to dance for a recruitment; the propagation and queuing delays of the path which a scout has traversed. Forooq's argument on this assumption is based on the estimation model, i.e., to estimate the trip time t_{in} required by a data packet to reach the next neighbouring node n from the current node i:

$$t_{in} \approx q_{in} + tx_{in} + pd_{in}$$

where q_{in} is the queuing delay for n at node i, and tx_{in} and pd_{in} are the transmission delay and the propagation delay of the link between them, respectively.

Additionally, q_{in} is dependent on the size of queue and the traffic load, tx_{in} depends on the size of the packet, and the bandwidth of the link between i and n, and pd_{in} models the delay that a packet experiences while travelling between the nodes [19].

The second observation from studying BeeHive is that as with the first one, the goodness of a neighbouring node j of a node l for reaching a destination d, is measured by a g_{jd}, a function of propagation and queuing delays, defined in BeeHive as:

$$g_{jd} = \frac{\frac{1}{p_{jd}}(e^{-\frac{q_{jd}}{p_{jd}}}) + \frac{1}{q_{jd}}(1 - e^{-\frac{q_{jd}}{p_{jd}}})}{\sum_{k=1}^{N} \frac{1}{p_{kd}}(e^{-\frac{q_{kd}}{p_{kd}}}) + \frac{1}{q_{kd}}(1 - e^{-\frac{q_{kd}}{p_{kd}}})}$$

Finally, what we can also observe is the way packet-switching is being performed by this algorithm. The method of selecting a neighbouring node as a next hop towards a particular destination is borrowed from the field of Genetic Programming (GP) [20]. Stochastic sampling with replacement (also known as roulette wheel selection method) ensures that a neighbouring node j with goodness g_{jd} will be selected as the next hop towards destination d at node i, with at least the probability of ϕ^i_{jd}, defined as:

$$\phi^i_{jd} = \frac{g_{jd}}{\sum_{k=1}^{N} g_{kd}}$$

Clearly, the selection probability of a node is predicated by its goodness, which in turn depends on the two delays discussed above.

What we strongly believe in our research is that, based on what we have learned from Biology and the rigorous study of honeybees in the literature, the bee-inspired network routing design can be explored further to achieve better results. Rather than considering only the propagation and queuing delays, our approach examines how different details that obviously play a significant role in Nature, can be directly mapped to the networks and used in a beneficial way. This requires up-to-date information to be collected and examined, by monitoring and evaluating different parameters of the communication network. The process involves cross-layer network design, which is described in the following section.

V. ADAPTATION USING CROSS-LAYER DESIGN

In this part of the paper, we give an introduction to the cross-layer design, and highlight three different routing protocols which illustrate how adaptation is achieved by combining knowledge from different layers. These protocols are just a sample of the relevant literature. However, they prove that cross-layer approach has a great potential in aiming our bee-inspired design.

A. Introduction and examples

The layered architecture simplifies development of different components by keeping each layer isolated from the others. Originated from the wired networks world, the concept of transparency is what makes OSI, TCP/IP and IEEE 802 models allow rapid and universal development and improvements.

Nevertheless, it has become evident that the traditional layered approach that separates routing, flow control, scheduling, and power control is suboptimal in the realm of wireless ad hoc networks. This can be attributed to the complex and unpredictable nature of the wireless medium. Thus, the need for adaptation in network protocols remains high. In order to tackle the problems faced in ad hoc networks, a cross-layer design [21] is desired to optimize across multiple layers of the stack. The basic idea of cross-layering is to make information produced or collected by a protocol available to the whole protocol stack, so as to enable optimization and improve network performance.

Until now several approaches have been proposed by researchers that use cross-layering in order to improve and optimize different network mechanisms. In most of the cases, the cross-layer design takes place between the media access control (MAC) and the physical (PHY) layers. However, there is a number of recent examples that illustrate the benefits of having other layers jointly designed, such as network-data link layer (DLL), or even application-network.

For instance, in order to bypass the resource constraints, Shah and Rabaey [22] have proposed an energy-aware routing protocol that uses a set of suboptimal paths occasionally to increase the lifetime of the network. The idea is that paths are chosen by means of a probability that depends on how low the energy consumption of each path is. The energy consumption is a result of signal strengths, a piece of knowledge that can be found at the MAC layer of the stack. Hence, cross-layering helped to access the information and use it to the network layer (routing layer) to make analogous decisions.

Another example, this time in link-aware routing was proposed by Lee and Gerla [23]. This protocol makes use of channel state information (CSI) [24] and cross-layer integration to route traffic along higher-capacity paths by consistently selecting channels with favourable conditions. This supports the idea that a node with multiple next-hop alternatives can measure the channel state on the links, and then forward a packet based on the link quality and other metrics.

Cross-layer has also been a great help in designing cost-aware routing approaches. Suhonen et al [25] have proposed a protocol that uses cost metrics to create gradients from a source to a destination node. The cost metrics consist of energy, node load, delay, and link reliability information that provide traffic differentiation by allowing choice among delay, reliability, and energy.

B. *Cross-layering in bee-inspired protocol engineering*

Before introducing cross-layer in bee-inspired protocol designing process, it is worth to briefly discuss what knowledge is held within the lower layers of the wireless network stack[2], and what channel, link, and transmission features can be explored.

Bearing in mind a packet's signal strength, in the PHY layer the received signal quality at a receiver is typically measured by the signal-to-interference ratio (SIR), which is the ratio of the power of the wanted signal to the total residue power of the unwanted signals. In order to correctly interpret the wanted signal, the SIR must be above a given threshold. For instance, a digital time division multiple access (TDMA) protocol at the MAC layer has a minimum acceptance SIR threshold is 14 dBm[3], whereas in power controlled multiple access (PCMA) protocol, it is only 6 dBm [26].

Bit error rate (BER) is a primary measure of data integrity at the PHY layer. It is an important reference for the assessment of any modulation scheme. BER can be defined as the estimated probability that any bit transmitted through the system will be received in error, e.g. a transmitted 1 will be received as a 0 and vice versa.

Furthermore, a wireless connection is prone to the background noise of the channel. If the strength of a transmission is significantly stronger than the noise, then the receiver is able to effectively ignore the noise and we say that the transmission has a good signal-to-noise ratio (SNR).

Coupled with this, another feature of the PHY layer is the packet error rate (PER). At the receiver side, the received signal is demodulated and bits are decoded and passed to the above layer on a packet-by-packet basis. After demodulation and decoding stages though, packets are received with some error probability called PER which depends on the post-detection SNR, the packet length, and current modulation and coding scheme [27]. A high PER results in repeated retransmission, and hence poor throughput.

The knowledge described above as well as the addressing information from the MAC layer can be collected and passed to the network layer in forms of encapsulated data or separate tables. Inspired by the factors of natural honeybees (presented in section III.A), we believe that this low-level knowledge can be used to estimate the fitness of a link, and the neighbouring node's integrity as well as reliability. We aim to develop a mechanism of representing them in such a way that honeybee agents will be able to make their decisions by interpreting a more comprehensive set of data.

VI. RESEARCH PLAN

After having built a satisfying level of knowledge by studying wireless network routing, both Internet-inspired and nature-inspired, and also understanding the mechanisms of cross-layering, we continue our research by following the work plan below:

- Clearly identify the attributes of the lower levels and their potential use in providing usable information.
- Analytically design our approach, combining nature-inspired concepts with the attributes above.
- Build an implementation of our design.
- Evaluate both design and implementation against the evaluation framework (will be described in section VIII).
- Compare our results with other approaches and publish the results.

VII. SIMULATOR ENVIRONMENT

For the implementation and testing of our approach as well as the comparison with existing methods, we plan to use the ns-2 network simulator [28]. Ns-2 is a packet-level simulator. It is a free object-oriented software package initially appeared as REAL in 1987. Nowadays it is part of the VINT project, collaboration between researchers at UC Berkeley, LBL, USC/ISI and Xerox PARC. It is classified as a centric discrete event scheduler that schedules events against time. A centric scheduler like the core of ns-2 cannot accurately emulate events that happen at the same time. Instead, events are handled one by one.

Furthermore, ns-2 project includes a number of other useful tools for logging, tracing and manipulating the results of any simulation. It is written in two programming languages, C++ and OTcl an object oriented version of Tcl. Taking advantage of the power and flexibility of these programming languages, ns developing process is based on the fact that all important parts of a network protocol are written in C++, and all the simulations are done by describing their scenarios with OTcl scripts.

Ns-2 network simulator has been one of the golden choices in research. People in both academia and industry use ns in

---

[2] Defined by IEEE 802.11 standards.

[3] All wireless power attributes are measured in decibel-meters (dBm), units that are related to milliwatts by the relationship dBm = 10xlog10(Watts / 0.001).

order to develop, improve and test various network-based issues from transport layer technologies such as TCP Vegas congestion control and TCP performance over Wireless Networks, to Bimodal Multicass and other transport layer protocols for Internet-compatible satellite-networks.

VIII. PERFORMANCE EVALUATION FRAMEWORK

An ideal routing algorithm should be able to route data packets as quickly as possible, and the amount of time it spends in processing the agents should be as small as possible.

In [29] such a prototype is introduced, which have been also used to evaluate BeeHive and other algorithms e.g. AntNet [30] and OSPF [31]. Parameters of this framework are the average throughput, the packet delay and delivery ratio, the session delay and completion ratio, the jitter, the control overhead, the total agent processing cycles per node as well as the data processing cycles per node, etc.

In order to both ensure the effectiveness and the performance of our approach, there is a need to run experiments subject to an evaluation framework. Hence, we need to take a detailed look at the above, and use it in order to satisfy the benchmark requirements.

IX. CONCLUSION

In this paper we presented our research position in the area of adaptive nature-inspired routing protocols for MANETs. We focus on bee-inspired network routing design and aim to enhance our approach with cross-layer information passed from low-level mechanisms of wireless communication. Our plan for the future is to design and implement these mechanisms in the ns-2 simulator, and compare the results with other state-of-the-art protocols.

REFERENCES

[1] C. E. Perins and P. Bhagwat, *Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers*. ACM SIGCOMM '94, 1994.

[2] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*. IETF RFC3626 http://www.ietf.org/rfc/rfc3626.txt, 2003.

[3] C. Adjin, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and D. Raffo, *Securing the OLSR protocol*. Proceeding of Med-Hoc-Net 2003, 2003.

[4] C. E. Perkins, E. M. Royer, and S. R. Das, *Ad hoc on demand distance vector routing*. IETF RFC3561 http://www.ietf.org/rfc/rfc3561.txt, 1999.

[5] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad-Hoc Wireless Networks*, pp. 153–81. Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer, 1996.

[6] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and systems*, ch. 6, pp. 80–95. Prentice-Hall, 2002.

[7] N. Beijar, *Zone Routing Protocol (ZRP)*. Licentiate course on Telecommunications Technology, Ad-Hoc Networking, 2002.

[8] M. Fleischer, *Foundations of Swarm Intelligence: From Principles to Practice*. Swarming Network Enabnled C4ISR, 2005.

[9] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

[10] G. D. Caro, F. Bucatella, and L. M. Gambardella, *Special Issue on Self-organisation in Mobile Networking: AntHocNet: an adaptive nature-insipred algorithm for routing in mobile ad hoc networks*. AEIT, 2005.

[11] M. Roth and S. Wicker, *Termite: Ad-Hoc Networking with Stigmergy*. IEEE, 2003.

[12] M. Roth and S. Wicker, *Termite: A Swarm Intelligent Routing Algorithm for Mobile Wireless Ad-Hoc Networks*. Cornell University, 2005.

[13] M. Saleem and M. Farooq, *BeeSensor: A Bee-Inspired Power Aware Routing Protocol for Wireless Sensor Networks*. Springer-Verlag, 2007.

[14] F. L. Lewis, *Wireless Sensor Networks*. To appear in Smart Environments: Technologies, Protocols, and Applications, ed. D.J. Cook and S.K. Das, John Wiley, New York, 2004.

[15] J. B. Free, *The Social Organization of Honeybees*. Edward Arnold, 1977.

[16] K. von Frisch, *The Dance Language and Orientation of Bees*. Oxford University Press, 1967.

[17] T. D. Seeley, *The wisdom of the hive: The Social Physiology of Honey Bee Colonies*. Harvard University Press, 1995.

[18] H. F. Wedde, M. Farooq, and Y. Zhang, *BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior*, pp. 83–94. M. Dorigo et al. (Eds.): ANTS 2004, LNCS 3172, Springer-Verlag Berlin Heidelberg, 2004.

[19] M. Farooq, *Bee-Inspired Protocol Engineering: From Nature to Networks*. Springer-Verlag Berlin Heidelberg, 2009.

[20] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[21] M. Ibnkahla, ed., *Adaptation and Cross Layer Design in Wireless Networks*. CRC Press, 2009.

[22] R. C. Shah and J. Rabaey, *Energy aware routing for low energy ad hoc sensor networks*, pp. 350–355. In Proceeding of the IEEE Wireless Communications and Networking Conference, Orlando, FL, 2002.

[23] S. J. Lee and M. Gerla, *AODV-BR: Backup routing in ad hoc networks*, pp. 1311–1316. In Proceeding of the IEEE Wireless Communications and Networking Conference, Chicago, 2000.

[24] A. I. Perez-Neira and M. R. Campalans, *Cross-Layer Resource Allocation in Wireless Communications*. Elsevier, 2009.

[25] J. Suhonen, M. Kuorilehto, M. Jannikainen, and T. D. Hamalainen, *Cost-aware dynamic routing protocol for wireless sensor networks design and prototype experiments*, pp. 1–5. In Proceeding of the 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland, 2006.

[26] J. P. Monks, V. Bharghavan, and W. mei W. Hwu, *A Power Controlled Multiple Access Protocol for Wireless Packet Networks*. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2001.

[27] T. Ue, S. Sampei, and N. Morinaga, *Symbol rate and modulation level controlled adaptive modulation/TDMA/TDD for personal communication systems*. Proc. IEEE Vehicular Technology and Communications (VTC), 1995.

[28] E. Altman and T. Jimenez, *NS Simulator for beginners*. ecture Notes 2003-2004, Univ. de Los Andes, Merida, Venezuela and ESSI, Sophia-Antipolis, France, 2003.

[29] H. F. Wedde and M. Farooq, *A performance evaluation framework for nature inspired routing algorithms*, pp. 136–146. In Applications of Evolutionary Computing, LNCS 3449, Springer Verla, 2005.

[30] G. D. Caro and M. Dorigo, *AntNet: Distributed Stigmergetic Control for Communications Networks"*, vol. 9, pp. 317–365. Journal of Artificial Intelligence Research (JAIR), 1998.

[31] J. Moy, *OSPF Version 2*. Internet Engineering Task Force. http://www.ietf.org/rfc/rfc2328.txt, 1998.