View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Aberystwyth Research Portal

Aberystwyth University

BeeIP: Bee-Inspired Protocol for Routing in Mobile Ad-Hoc Networks

Wilson, Myra S.; Giagkos, Alexandros

Publication date:
2010

Citation for published version (APA):
Wilson, M. S., & Giagkos, A. (2010). *BeeIP: Bee-Inspired Protocol for Routing in Mobile Ad-Hoc Networks*. 263-272. http://hdl.handle.net/2160/5298

General rights
Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

Download date: 30. Aug. 2021

BeeIP: Bee-Inspired Protocol for Routing in Mobile Ad-Hoc Networks

Alexandros Giagkos and Myra S. Wilson

Dept. of Computer Science, Aberystwyth University, Penglais, SY23 3BD, Wales, UK
`{aag07,mxw}@aber.ac.uk`

Abstract. We introduce a new bee-inspired routing protocol for mobile ad hoc networks. Emphasis is given to the ability of bees to evaluate paths by considering several quality factors. In order to achieve similar behaviour in the networking environment, BeeIP is using cross-layering. Fetching parameters from the lower PHY and MAC layers to the core of the protocol, offers the artificial bees the ability to make predictions about the link's future performance. Our approach is compared with two well-known routing protocols in the area, the destination sequenced distance-vector protocol (DSDV), and the adaptive on-demand distance vector protocol (AODV). The outcome shows that BeeIP achieves higher data delivery rates and less control overhead than DSDV, and slightly better results compared to AODV, initializing less route discovery processes.

Key words: Bee-Inspired, Network, Routing, Cross-Layer, MANETs.

1 Introduction

Nodes in a mobile ad hoc environment face two major challenges, the mobility of the network participants and their resource constraints. Firstly, the movement of the nodes lead to network topology changes and frequent path breaks. Secondly, as nodes act as both transmission endpoints and routers, they generate their own traffic as well as route the traffic generated by others. This requires more energy being spent, and also, increases the complexity of routing. [1]

The routing algorithms for mobile ad hoc networks (MANETs) can be broadly categorized as proactive, reactive or hybrid. Protocols that use proactive algorithms periodically send control packets to collect information about the network state and update their routing tables accordingly. Such examples are the destination sequenced distance-vector protocol (DSDV) [2] and the optimized link state routing protocol (OLSR) [3].

Contrarily, reactive algorithms find routes on-demand. They do not maintain routes between all the nodes in the topology. Rather, routes are established only when needed through a route discovery process, in which a route request is broadcast. Examples of reactive protocols are the dynamic source routing (DSR) [4] and the adaptive on-demand distance vector (AODV) [5].

The third category, hybrid, contains ideas borrowed from both proactive and reactive paradigms. Generally, hybrid protocols separate the network topology into zones. Routing is determined proactively within each zone, and reactively outside it. The 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) [6].

All these approaches point out the need for adaptation in routing. Protocols have to be able to adapt to topological changes and provide optimal results. Examples of such adaptive behaviour come from the study of Nature and in particular natural networks (e.g. insects). The first algorithm which presented a detailed scheme for network routing based on ant colony principles is ARA [7]. This routing algorithm is inspired by the pheromone laying behaviour of ant colonies.

In this paper we present a new routing protocol for MANETs, called BeeIP, which is designed to provide routing solutions inspired by the foraging principles of bees. Cross-layering is used in order to utilize parameters of lower layers and be able to calculate the performance of the links between the sources and the destinations [8].

2 Related Work

In 2004, H.F. Wedde, M. Farooq, and Y. Zhang were the first to present BeeHive [9], a novel routing algorithm for wireless networks inspired by the communicative and evaluative methods and procedures of bees.

More specifically, BeeHive is built around two types of agents, the short distance and the long distance agents which are proactively generated at the nodes and are designed after the way bee foragers respond to bee dances. The responsibility of both types of agents is to explore the network and to evaluate the quality of paths that they traverse, in order to update node routing tables. Short distance agents are allowed to move only up to a restricted number of hops in the network, whereas long distance agents have to collect and disseminate routing information in the complete topology.

Moreover, 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, bee agents occupy smaller bandwidth and require significantly less processing time compared to the agents of existing algorithms.

BeeHive has been an inspiration to further research and enhancements. In 2005, H.F. Wedde et al have proposed BeeAdHoc [10], a routing algorithm for energy efficient routing in MANETs. By utilizing two types of agents, scouts and foragers, BeeAdHoc is able to reactively search for routing solutions, consuming less energy compared to existing state-of-the-art approaches. The major difference of our approach and BeeAdHoc is on how the quality of the links is calculated, and the way of evaluating their performance. Due to the early stage of this work, we were not able to present a comparison between our approach and BeeAdHoc, however, this is part of our future plan.

The rest of this paper is organized as follows. In Section 3 we give an overview of the key points of biological bee behaviour in respect to both scouting and foraging. In Section 4 we present our design model. Section 5 includes the first simulation experiments and results. Section 6 contains our conclusion and plan for future work.

3 Biological background

In Nature, a bee explores the surroundings of the hive in order to detect possible sources of food. Once a source is found, the scout 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.

In his book [11] von Frisch presented the understanding of the dependence of the bee dances on the profitability of foraging activity. He has shown that although the pattern of bees' dance is determined fundamentally by the distance of and direction to a source of food, whether dancing will take place depends on many factors that may significantly change the bees' behaviour. Examples of such factors are the sweetness of the sugar solution in the food, the ease of obtaining and carrying it back to the hive, the distance of the food source to the hive, and the amount of energy required during the particular foraging process.

It is also crucial to mention that the special dance is not performed only by scout bees. Each time a successful forager returns back to the hive she can also perform the foraging dance (serving as a scout at the same time), and report any improvement or deterioration of the currently working path. Furthermore, if the path's reliability is becoming very poor, the forager can also refuse to dance and, hence, stop recruiting new members.

4 Design model

BeeIP is a routing protocol which models the collaborative behaviour of simple artificial bee agents to build enough knowledge in order to establish communication links between two nodes, and allow data to be transferred across them.

The base of any assumptions made in our design is that every time there is a need for a link to be established, the sender node will behave as being the hive, the destination node will behave as being the source of food, and all the intermediate nodes will consistite the path that a bee forager needs to traverse while flying from one endpoint to the other.

4.1 Agents

The model uses three types of agents in the form of data packets. The scout, the ack_scout, and the forager.

Scout: They are sent when a scouting process is initialized in order to discover new paths towards a given destination. This happens each time there is a new request from the upper layer and previous routing knowledge is unsatisfactory. A scout is transmitted using broadcast to all neighbouring nodes. This technique benefits not only the propagation of the initial request, but also the introduction of the transmitting node to its neighbourhood.

Apart from the details of the scouting process, scouts also carry important information about their sender's state. A node's state is a group of attributes that describe the situation in which the node is at the time of broadcasting the scout packet. Cross-layering between PHY, MAC and network layers allows the routing protocol to know the current energy and speed levels of the node, as well as the size of the interface queue.

Furthermore, upon receiving a scout, neighbouring nodes are able to discover evidence about the link's quality between them and the scout's sender. This evidence is the one-way transmission delay of the link, and the scout packet's signal power. The latter is an indication of the distance and the clearance of the intermediate area. The information above is stored internally and is used to calculate the local reliability level of the pair, i.e. the sender and the receiver of the scout packet. The local reliability level plays a very important role to the overall path quality and the decision making of the foragers.

Following that, the receiving node can either propagate the scout packet further if it is not the scout's destination, or create an ack_scout to send back. Loops are avoided by tagging each scout packet with a unique scouting ID.

Ack_scout: Once the scout reaches its destination the scouting is considered successful and an ack_scout packet is created. Ack_scouts use a source routing fashion to travel back to the source, using unicast transmission. Therefore, the route that was followed towards the destination is used in reverse. On their way back, ack_scouts acknowledge the success of the scouting to both the intermediate nodes and the source node.

Forager: When BeeIP is unable to transmit a data packet, it stores it into a local queue and starts a new scouting process for its destination. This decreases the packet loss due to incomplete routing information. Once an ack_scout returns back and acknowledges the existence of a path, all packets for the corresponding destination in the queue are being transmitted.

The way they do this, is by using the most important agent type of BeeIP, the forager. Foragers are specially crafted packets that have three important roles. Firstly, they carry (in form of payload) the data packets from the source to the destination. Secondly, they are used to update neighbouring nodes' states and links' information, just like scouts did in the first place. Thirdly, foragers are constantly monitoring the path they traverse for any improvements. Technically speaking, foragers collect the differences between the local reliability levels, calculated by using the current forager, and the local reliability levels calculated by the previous forager's visit, and report the summation back to the hive. In a

TCP connection, this is done when carrying TCP ACK packets. The summation represents the total reliability level of the path, hence, the global reliability level.

4.2 Local reliability level

The local reliability level describes the one-way performance between a pair of nodes in the topology. It is the BeeIP's way of measuring how good or bad a transmission can be, by using this particular pair. The direction of the measurement is towards the source of the transmission and it is a combination of the neighbour's state and the network link between the endpoints.

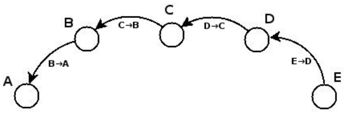

Fig. 1. *A simple scouting.*

Figure 1 illustrates a simple example. Node A is the source (hive), node E is the destination (flower), and nodes B, C and D consistute the intermediate path from A to E. A forager that returns back to A has to follow the path through the nodes D, C, B, and A, pick up each pair's reliability improvement, and submit the result to A. Furthermore, the reliability level of pair (E,D) is a combination of E's state, and the path $E \rightarrow D$.

In total, there are five parameters that can be extracted from a node's state and a path such as the above. (i) E's speed level, (ii) E's energy level, (iii) the path's transmission delay based on the forager packet, (iv) the forager packet's transmission power, and (v) the queuing delay of the path, based on E's reported queue size.

Each one of the above parameter plays a significant role in the local quality of the pair. The speed of a node affects their transmissions and can lead to a weak or even broken link. Similarly, the energy level of the node dominates its ability to transmit clear signals to full transmission range. The signal power of the orager packet is used to give an idea for both the distance and the area between the nodes. Finally, both queueing and transmission delays alter the quality of the link. On one hand, the transmission delay describes the difficulties experienced because of the bandwidth of the link. On the other hand, the queueing delay describes the difficulties caused by traffic loads. Note that the propagation and processing delays are factored out, since they are insignificantly small.

Although the parameters' similarity is that they all affect the reliability level of the pair, their values are of different scales. Table 1 shows the minimum and maximum accepted values for each parameter during the simulation experiments. In order to use them properly, all values need to be normalized to the same scale: min 0, max 20. BeeIP achieves scale normalization of values by performing linear transformation. If α_1, β_1 and α_2, β_2 the minimum and maximum numbers of the first and second scale respectively, and χ is the number to be normalized to ψ then,

$$\psi = \frac{\alpha_2 + (\chi - \alpha_1) * (\beta_2 - \alpha_2)}{(\beta_1 - \alpha_1)} \tag{1}$$

Notice that speed, and both queueing and transmission delays are adversely affecting the performance. For instance, a node's speed equal to 0 does not affect the transmission, as it does not alter the distance between the transmission endpoints. In order to tackle this issue, these three parameters are normalized in reverse.

	Signal Pow[1]	Speed	Energy	Q-Delay[2]	Tx-Delay[3]
min	1.258925e-10 W (-69 dBm)	0 m/s	0 W*h	0 s	0.0006 s
max	7.943282e-10 W (-61 dBm)	10 m/s	10 W*h	0.075 s	0.0120 s

Table 1. *Local reliability parameters and scales.*

Once all values are put on the same scale, the local reliability level is calculated using a simple weighting system. This is required because not all of the parameters have the same influence on the performance. Obviously, a very weak signal strength can be an indication of either a long distance between two nodes or the appearance of an obstacle. In both cases, it requires immediate action. This does not happen with the queueing delay. The latter may affect the performance, however, it does not necessarily involve a link break. The weighting system is shown at Table 2.

Parameter:	Signal Pow	Speed	Energy	Q-Delay	Tx-Delay
Weight (w):	0.40	0.20	0.20	0.15	0.05

Table 2. *Weighting system and factors.*

Then, the local reliability level of the pair is finally defined by the formula:

$$rel_{local} = pow' * w_{pow} + speed' * w_{speed} + energy' * w_{energy} + qd' * w_{qd} + txd' * w_{txd} \tag{2}$$

where pow' is the normalized value of the signal's power, etc.

Every time a forager visits a new node during its flight back to the hive, the knowledge it brings with it as well as its own transmission are used to calculate the new local reliability level of the corresponding pair. Once calculated, the number is compared with the previous available local reliability level. The difference of the two is then reported back to the forager which continues its journey to the next hop in the path.

The difference of the two local reliability levels, previous and new, describes the improvement of the pair since the last use. In addition, the new local reliability level is stored internally to be used for future calculations.

---

[1] Proxim. ORiNOCO 11b Client PC Card Specification for open range environment.

[2] Maximum queue size is set to 50 packets.

[3] 11Mbit bandwidth. Minimum 76 bytes and maximum 1500 bytes packet size.

4.3 Global reliability level

A bee forager that finally arrives at its hive, carries the summation of all the local reliability differences collected on its way back. This number is called the global reliability level and is an indication of the link's quality as experienced during the last forager's flight. In BeeIP it is defined as follows:

$$rel_{global} = \sum_{n=1}^{m} (rel_{local-new_{N_{n+1} \to N_n}} - rel_{local-prev_{N_{n+1} \to N_n}}) \tag{3}$$

where m is the total number of nodes in an numerically ordered path, and $N_{n+1} \to N_n$ the pair of nodes with direction towards the source node (N_1).

Likewise in local, the global reliability level is compared to the one obtained from the previous flight. The difference of the two represents the improvement or the deterioration of the quality of the path. However, the number by itself can only give a dim idea since it is a result of one transmission only, which, depending on the environmental and network conditions may lead to negative assumptions. In order to utilize these numbers correctly and be able to make predictions about the quality of the link and its status in future, we use a 10x2 matrix of the last 10 instances of incoming foragers and apply regression analysis to the values. Time is used for the first column, and the difference of the new and previous global reliability levels for the second column. The output matrix has the form of:

$$\begin{pmatrix} 2.823042 & 0.32 \\ 2.825661 & 1.46 \\ \multicolumn{2}{c}{..} \\ 2.854530 & -0.25 \end{pmatrix}$$

Using Pearson's correlation coefficient [12], we are allowed to make predictions based on the strength of the linear dependence between the two. The correlation coefficient r is defined by the formula:

$$r = \frac{\sum_{i=1}^{k} (t_i - \mu t)(rel_{global_i} - \mu rel_{global})}{\sqrt{\sum_{i=1}^{k} (t_i - \mu t)^2} \sqrt{\sum_{i=1}^{k} (rel_{global_i} - \mu rel_{global})^2}} \tag{4}$$

where t_i the time of receiving rel_{global_i}, μt the mean of the time column values, and k the matrix row number (10 by default).

The correlation coefficient result ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between t's and global reliability differences perfectly, i.e. when t increases, the improvement increases too. On the contrary, a value of -1 implies that the improvement decreases as t increases, i.e. the path becomes weak. Values near 0 imply that there is no linear correlation between the two, and we are not able to make any serious predictions.

Similarly to Nature, where bee foragers may dance vigorously if the quality of the path is becoming better or even stop dancing when the path is very poor, artificial foragers are able to judge whether to recruit other members or initialize a new scouting process. At this early stage of this work, BeeIP is able to detect weak links by comparing r to a threshold (-0.8) and re-send new scouts if it finds it necessary.

5 Simulation experiments and results

In order to evaluate the performance of BeeIP, we have used ns-2 network simulator. We have performed experiments with static scenarios of 20, 40, 60, 80 and 100 nodes in 300x300 m^2, 500x00 m^2, .., and 1100x1100 m^2 areas. Each node carries a single wireless card, the configuration of which is set to match ORiNOCO11b Wireless Card, 11Mbps, 802.11b for 160m in open range environment.

The nodes are moving in random directions with randomly selected speeds between 1m/s (walking speed) and maximum 10m/s. Two nodes, fairly far from each other, are picked up to serve as the source (bee hive) and the destination (flower) of an TCP/FTP connection in each scenario. The initial energy level is set to 36000 Joules (or 10 watt-hours).[1] The simulation time is set to 600 seconds. Our results are compared to those of AODV and DSDV protocols, under the same topological conditions. In order to factor out any implementation related errors to our comparisons, we use the implementations which are distributed with ns-2 simulator.

In figure 2, we study the successful packet delivery ratio of the three protocols. Unsurprisingly, the reactive nature of BeeIP has a clear advantage over DSDV which becomes weaker as the number of nodes is increased. This is due to the large number of required control packets, in order to collect enough information and build DSDV routing tables. Compared to AODV, BeeIP has a slightly better performance. The reason behind that, is that although they both apply reactive schemes, BeeIP is able to detect when a link is about to break faster, and then switches to another one.

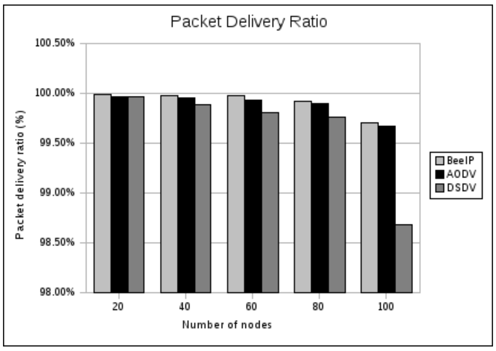

Fig. 2. *Packet delivery ratio vs. number of nodes.*

Furthermore, we have measured the packet loss of the three approaches (figure 3). For a number of nodes lower than 40, BeeIP scores less packet loss than AODV. Between 40 and 60 nodes, BeeIP performs quite steadily and better than AODV. However, we notice a big increase after 60 nodes, which although is still better than AODV's, it triggers our interest for future improvements. Finally, both BeeIP and AODV packet losses are significantly lower than DSDV's.

---

[1] Almost $^1/_{25}$ of the battery capacity of a fully charged Pioneer P3-AT all-terrain robot (http://www.activrobots.com/ROBOTS/specs.html).

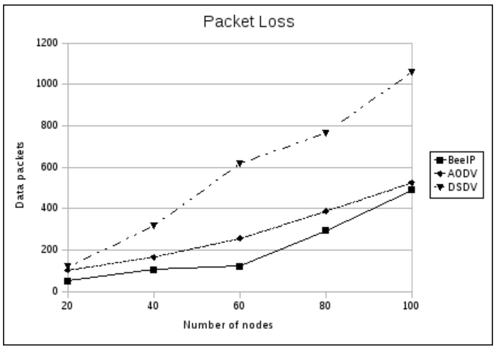

Fig. 3. *Packet loss vs. number of nodes.*

The control overhead of the three protocols is shown in figure 4. Although things get worse for higher number of nodes, under the same circumstances, BeeIP sends less control packets during the static simulation scenario, than AODV and DSDV protocols.

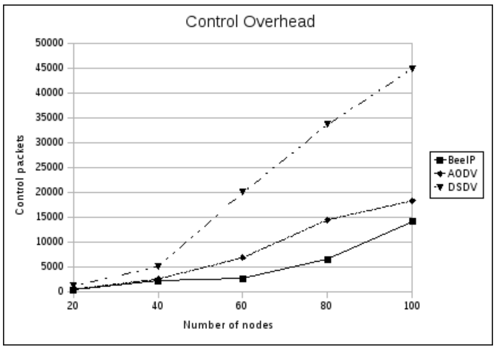

Fig. 4. *Control overhead vs. number of nodes.*

In a reactive point of view, BeeIP and AODV are compared based on the successful route discoveries and the successful data packet deliveries. All experiments have shown that BeeIP managed to deliver more data packets successfully using less route discoveries. Table 3 summarizes these results. For example, BeeIP was able to send 369399 data packets by using 14 links during the simulation, whereas AODV sent 310912 using 15 links.

	20	40	60	80	100
BeeIP:	14 (369399)	27 (333449)	146 (381166)	184 (355629)	271 (163024)
AODV:	15 (310912)	39 (330284)	196 (374010)	220 (354899)	343 (157042)

Table 3. *Successfully established links during simulation (packets sent).*

6 Conclusion

In this paper we have introduced BeeIP, a new bee-inspired routing protocol for mobile ad hoc networks. We have also compared the first simulation results of our approach with two state-of-the-art protocols, AODV and DSDV. The simulation experiments have shown that BeeIP performs better than DSDV and slightly better than AODV in terms of packet delivery ratio and packet loss. Furthermore, BeeIP was able to deliver more data packets successfully, initializing less route discovery processes than AODV under the same network conditions.

Our future work includes the improvement of our design, in order to support multiple paths for each transmission which will be selected based on their quality via artificial bee dancing. This will increase the life of the network and the delivery ratio of the protocol. Finally, we need to add support for stateless transport protocols such as UDP. The results of these improvements as well as new features, will be compared to AODV, DSDV and the biologically inspired AntHocNet and BeeAdHoc, something that we did not include in this paper due to the early stage of the work.

References

1. Murthy, C., Siva, R., Manoj, B.S.: Ad Hoc Wireless Networks Architectures and Protocols. Prentice Hall (2004)
2. Perins, C.E., Bhagwat, P.: Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers. ACM SIGCOMM '94 (1994)
3. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). IETF RFC3626, http://www.ietf.org/rfc/rfc3626.txt
4. Johnson, B.D., Maltz, D.A.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Mobile Computing, T. Imielinski and H. Korth (eds.), pp. 153–81. Kluwer (1996)
5. Perkins, C.E., Royer, E.M., Das, S.R.: Ad hoc on demand distance vector routing. IETF RFC3561, http://www.ietf.org/rfc/rfc3561.txt
6. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). IETF RFC3626, http://www.ietf.org/rfc/rfc3626.txt
7. Gunes, M., Sorges, U., Bouazizi, I.: ARA - The Ant-Colony Based Routing Algorithms for MANETs. IWAHN (2002)
8. Giagkos, A., Wilson M.S.: A Cross-Layer Design for Bee-Inspired Routing Protocols in MANETs. In: Proceedings of TAROS 2009, Kyriacou, T., et al (eds.). TAROS 2009, pp. 25–32. Ulster University (2009)
9. Wedde, H.F., Farooq, M., Zhang, Y.: BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior. In: Dorigo, M. et al. (eds.). ANTS 2004. LNCS, vol. 3172, pp. 83–94. Springer (2004)
10. Wedde, H.F., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller C., Meth, J., Jeruschkat, R.: BeeAdHoc: An Energy Efficient Routing Algorithm for Mobile Ad Hoc Networks Inspired by Bee Behavior. In: GECCO. ACM (2005)
11. Frisch, von K.: The Dance Language and Orientation of Bees. Oxford University Press (1967)
12. Read, T.R.C, Noel A. C. Cressie, N.A.C.: Goodness-of-fit statistics for discrete multivariate data. Springer (1988)