BeeIP

Giagkos, Alexandros; Wilson, Myra Scott

Published in:
Information Sciences

DOI:
10.1016/j.ins.2013.12.038

Publication date:
2014

Citation for published version (APA):
Giagkos, A., & Wilson, M. S. (2014). BeeIP: A Swarm Intelligence based routing for wireless ad hoc networks. *Information Sciences*, *265*, 23-35. https://doi.org/10.1016/j.ins.2013.12.038

Information Sciences 265 (2014) 23–35

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BeeIP – A Swarm Intelligence based routing for wireless ad hoc networks

Alexandros Giagkos, Myra S. Wilson *

Dept. of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, UK

A R T I C L E I N F O

Article history:
Received 22 February 2013
Received in revised form 1 November 2013
Accepted 29 December 2013
Available online 7 January 2014

Keywords:
Swarm intelligence
Wireless
Mobile ad hoc network
Sensor network
Ant-inspired
Bee-inspired

A B S T R A C T

Agent-based routing in wireless ad hoc networks defines a set of rules that all the participating nodes follow. Routing becomes a collaboration between nodes, reducing computational and resource costs. Swarm Intelligence uses agent-like entities from insect societies as a metaphor to solve the routing problem. Certain insects exchange information about their activities and the environment in which they operate in order to complete their tasks in an adaptive, efficient and scalable manner. This paper examines Swarm Intelligence based routing protocols, along with a newly proposed bee-inspired routing protocol for providing multi-path routing in wireless ad hoc networks of mobile nodes. Simulation results indicate that applying Swarm Intelligence offers a significant level of adaptability and efficiency that, under several network conditions, allow the protocol to outperform traditional approaches.

1. Introduction

The constant improvement of the technologies related to telecommunication and computer networks is one of the fastest growing aspects of people's needs. The Internet has revolutionized many aspects of daily life. In fact, it has created the user need and demand to be connected any time and anywhere. Wireless communication networks have played a critical role in fulfilling those telecommunication needs.

Wireless ad hoc networks are those networks which have no fixed infrastructure. As opposed to the infrastructure counterparts, where wireless access points and routers are present, in an ad hoc environment there is neither an organised hierarchy nor central administration to orchestrate the data traffic between the participating devices (nodes). Rather, the routing problem is meant to be solved by the nodes themselves, in a decentralized and distributed manner and can be defined as the construction and maintenance of a working path or paths between two nodes in a network, which wish to communicate with each other by exchanging data packets.

Solving the complex routing problem is the subject of active research. The literature lists a plethora of protocols which apply routing strategies inspired by the traditional wired Internet. Although those mainstream approaches are widely accepted as the state-of-the-art solutions, they require the routing protocol to forward data packets to the next hop in the path based on topological information collected by non-intelligent mechanisms. While the user needs change and increase, the cost of such approaches in terms of resource and processing power is also increased. Due to this fact, the research community has turned its attention to a different approach, proposing agent-based networking systems.

Swarm Intelligence (SI) has come to represent the idea that it is possible to control and manage complex systems of agent-like entities, which, based on their multiple interactions with the environment and with each other, are able to provide

---

* Corresponding author. Tel.: +44 1970 622928.
E-mail addresses: alg25@aber.ac.uk (A. Giagkos), mxw@aber.ac.uk (M.S. Wilson).

0020-0255/$ - see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.038

solutions to difficult problems. The underlying features of SI are based on observations of social insects. For instance, ant colonies and bee hives are populated by small creatures which, despite their size are capable of finding and storing food, producing goods such as honey, wax or propolis, mating, protecting their young and guarding the nest.

This paper focuses on the agent-based systems that are found in insect societies and are described by Swarm Intelligence. It discusses their approaches and applications in wireless networks with mobile and ad hoc characteristics. In addition, BeeIP, a newly proposed SI-based routing protocol for MANETs based on adaptive honeybee foraging is presented. The rest of the paper is as follows. Section 2 describes the types of wireless networks under review and defines the routing problem and its challenges in such networks. In Section 3, recent milestones of research in the area of SI-based routing protocols are presented and classified according to their mechanisms. BeeIP is presented in Section 4, followed by a qualitative and a quantitative comparison to related honeybee-inspired work and the state-of-the-art protocols respectively. Finally, the paper concludes with a summary of the results and a discussion of future research paths with respect to real life applications.

2. Mobile ad hoc networks and wireless sensor networks

The current literature focuses on two major types of wireless ad hoc networks, the Mobile Ad hoc Networks (MANETs) [45,44] and Wireless Sensor Networks (WSNs) [1,74].

In a MANET, all the participants (nodes) are mobile and use only wireless communication links to send and receive data packets. Self-organisation, configuration and maintenance are built into the network, in the form of network protocols which need to be highly dynamic and operate in a fully distributed way. They also need to be adaptive to any topological changes and robust in order to face unreliable links between transmitting nodes. Finally, due to the network limited resources, protocols need to provide cheap solutions in terms of energy consumption, bandwidth and computing power.

Wireless Sensor Networks (WSNs) are ad hoc networks which consist of a large number of nodes equipped with some sort of sensor and are generally deployed in order to measure physical parameters of a certain phenomenon. Network protocols designed for WSNs need to be robust and self-configured. Due to the lightweight and cheap nature of the sensor nodes, protocols need to be able to operate using as little resources as possible for data processing, storage and transmission power, and need to scale up easily without losing efficiency.

The two major challenges for the above systems are node mobility [54,20] and limited energy resources [66]. The impact node mobility has on a routing protocol's performance is an established research problem and can mainly be seen as the product of frequent link breaks, energy consumption and changes to the node density within the topology. Refs. [7,68] show that the higher the density of the network, the greater the load within it. Throughput is affected as more control overhead is broadcast to the network and the greater the node density, the greater the packet loss rate. Clustering is also a frequent problem with node mobility where traffic is routed through the nodes at the group edges, causing bottlenecks and load congestion.

Nodes consume energy while sending, receiving and even discarding packets. Studies have shown that memory allocation at the mobile node consumes a high level of power [2]. Route discovery, routing maintenance and topology self-organisation require extra control packets, thus spending more energy. Power consumption in wireless ad hoc networks is directly proportional to route length.

Routing protocols for wireless ad hoc networks are traditionally classified into several types (proactive, reactive, or hybrid), based on their routing approach and mechanisms [40,9]. In proactive or table-driven routing protocols, the network topology information is maintained within each node in the form of routing tables, which are periodically exchanged in order for the information to be kept up-to-date. Destination Sequenced Distance-Vector (DSDV) [51], a proactive routing protocol, is considered to be the de facto standard in the area of proactive routing in MANETs. Although proactive routing is often seen as not the best approach for wireless ad hoc networks, especially under large number of nodes and high mobility rates [32], DSDV is heavily studied and used in protocol comparisons in the literature.

Unlike proactive protocols, reactive (also called on-demand) protocols do not maintain routes between all the nodes in the wireless ad hoc network. Rather, routes are established when needed through a route discovery process in which a route request message is broadcast. A route reply is returned either by the destination or by an intermediate node with an available route. Ad hoc On-Demand Distance-Vector protocol (AODV) [52], is a reactive routing protocol which is heavily studied in the literature. AODV's performance is generally considered to set the benchmark standards in evaluating routing protocols for MANETs, as the protocol participates in most of the existing work in this research area. This is also due to the current experimental status of its draft by the IETF MANET group, which sets AODV under investigation for standardization. Ad hoc On-Demand Multi-path Distance-Vector (AOMDV) [42] is a multi-path extension to AODV. Dynamic Source Routing (DSR) [35], also a reactive routing protocol, has gained considerable attention because of its ability to dramatically reduce control overhead, even under high rates of node mobility. DSR is similar to AODV in terms of route discovery, however it uses source routing in order to forward packets. Source routing reduces the computation complexity of deciding the next hop, as well as the amount of storage required to keep routing information, at each intermediate node.

A hybrid protocol in wireless ad hoc networks, such as the Zone Routing protocol [5] is an adaptive routing protocol that combines the best features from both reactive and proactive techniques. 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 optimisation within the zones.

3. Swarm intelligence

Swarm Intelligence is the field of study where solutions to similar search problems are given by utilising agent-based systems inspired by the social behaviour of natural systems, mostly found in insects. The discussion of the characteristics and the challenges found in wireless ad hoc networks brings to light a number of needs that are treated by the collective behaviour described by SI. The distribution of the computational costs and the cooperation between the agents allows the system to be self-organised, robust and achieve tasks that could not be done by a single agent at a global level. The individuals must combine their efforts in order to successfully solve a problem that goes beyond their individual capabilities. Following the paradigm of division of labour and changing roles, routing protocol agents can dynamically change their behaviour according to the different states of solving the routing problem, whereas at the same time require a less complex design [30]. Being distributed and able to switch roles and allocate tasks in an adaptive manner allows the system not only to be robust and fault tolerant, but also to be scalable in terms of agents' population, work load and area coverage. Additionally, scalability allows agents to join and leave the system at any time, with no serious effect at a global level.

Examples of solving metaphors of the problem are met in ants and honeybees, and in particular, their ability to search for food sources and balance the transportation costs with the productivity of the colony. The means of communication changes according to the species. For instance, worker ants choose paths between the nest and interesting sources of food based on the concentration of pheromone on each path. Honeybees are able to find interesting flowers, exchange information about them and adapt their foraging accordingly, by performing their special dance.

Under the umbrella term of Swarm Intelligence there are two main meta-heuristics of research interest in connection to designing network routing protocols: ant colony optimisation and bee colony optimisation.

3.1. Ant colony optimisation

The Ant Colony Optimisation (ACO) [19] meta-heuristic has been inspired by the operating principles of ants, which allow them to perform complex tasks such as foraging and nest building.

Based on the same idea as AntNet [16] for wired networks, Gunes et al. proposed ARA [31], a purely reactive protocol. ARA consists of three phases. The route discovery phase is when new routes are created by the use of forward and backward ants. A forward ant is responsible for establishing the pheromone track to the source node and a backward ant establishes the pheromone track to the destination. The second phase of ARA is the route maintenance. The third phase of ARA is called route failure handling where a broken link is detected by a missing acknowledgement packet. In terms of performance, simulation experiments have shown that ARA is very close to the performance of the state-of-the-art Distributed Source Routing (DSR) [35], but generates less overhead [31].

The Termite protocol [58] takes into consideration the ability of social insects to self organise and is based on the concept of the termite hill building [59] behaviour. 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.

AntHocNet is a hybrid algorithm proposed by Ducatelle [21], Di Caro et al. [18]. If a session does not have up-to-date routing information, it reactively sends out an ant-like agent, called reactive forward ant, in order to look for paths to the destination. 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 a backward ant. Simulation experiments have shown that AntHocNet can outperform Ad Hoc On-Demand Distance-Vector Routing (AODV) (a de facto reactive routing protocol) in terms of delivery ratio and average delay [21].

Proposed by Selvakennedy et al. [64], T-ANT, provides clustering, routing and data gathering in WSNs. It uses the ACO meta-heuristic in order to aid cluster head election in such a way that optimal data aggregation with minimised overall energy dissipation can be achieved. The authors of T-ANT have conducted comparison experiments to other related work, such as LEACH [33] and TCCA [63], the results of which show that T-ANT is able to store less state overhead in memory, providing decentralized and robust clustering with no position knowledge of the sensor nodes.

Proposed by Camilo et al. [13], the Energy Efficient Ant-Based Routing protocol (EEABR) is an enhancement to the ant colony optimisation meta-heuristic in order to save energy in WSNs. After simulation experiments, it has been shown that EEABR's special ants minimise communication loads and maximise energy savings, contributing to expand the lifetime of the wireless network.

Hybrid in nature, the HOPNET [70] protocol divides the network into zones, each one defining the neighbourhood of a node. Each node keeps routing information for paths to nodes within its neighbourhood (intrazone routing) and for paths to nodes outside its zone (interzone routing). Comparison experiments indicate that HOPNET is quite stable for high and low mobility networks and is highly scalable as the network size has no impact on the protocol's performance.

3.2. Bee colony optimisation

Similar to ACO, the Bee Colony Optimisation (BCO) [67,41] is a nature-inspired meta-heuristic, which can be applied in order to find solutions for difficult combinatorial optimisation problems. Once the problem is defined, the forward pass starts by artificial bees seeking a partial solution to the problem. After finding enough information, they return to the hive where the partial solutions they have obtained are shared between the searchers. Each artificial bee has to make a decision based on a certain probability, whether it will continue searching following its own path, or switch to a fellow searcher's solution. The final step is to decide which solution is the best, based on certain criteria.

BeeAdHoc [72] was designed with the aim of creating an energy efficient routing protocol. Initially, a packer agent represents a food-storer bee that resides inside the network node. BeeAdHoc uses scout agents to discover routes between sources and destinations. In a reactive way, when a route is required at the destination node, a scout agent is broadcast to the network just like a forward ant in ant-inspired protocols. Once a scout returns to its source node, it recruits foragers by using a metaphor of bee dance, as bees do in nature. Finally, a forager is the bee agent that receives data packets from a packer and delivers them to their destination, in a source-routed modality [24,53]. The authors show that BeeAdHoc consumes significantly less wireless network card energy as compared with DSR, AODV, and Destination-Sequenced Distance Vector (DSDV) [51].

BeeSensor [60] focuses on minimising the energy costs using bee agents; the packers, scouts and foragers. As in BeeAdHoc, packers are located within the nodes. At the sensor nodes their purpose is to receive data from the transport layer and load them to an appropriate forager. At the sinks, packers recover data from the incoming foragers and deliver them to the transport layer. Route discovery is achieved by using forward scouts and backward scouts. Results show that the honeybee-inspired protocol is able to transmit more packets than an energy optimised version of AODV [61], achieving less control overhead and lower energy consumption.

In Stigmergic Landmark Routing [38], reactive packets are sent to discover new paths. When the destination is received, the packets retrace their path, and while travelling back to the source, store part of it in key nodes. At the source node, a virtual dance is performed by which search experiences are advertised to other members. SLR is considered a hybrid protocol as its route discovery packets can change their behaviour and proactively follow previous information (take previously discovered routes) at an intermediate node, while exploring the topology to build a route between the source and the destination. Lemmens and Tuyls provide a quantitative comparison to AODV, followed by a qualitative one to AntHocNet routing protocol. The results show that SLR is able to outperform AODV in networks smaller than 100 nodes, in terms of packet delivery ratio. Due to the lack of source code availability for the ns-2 network simulator, a direct quantitative comparison to AntHocNet is not present. However, after a qualitative indirect comparison the authors suggest that SLR may perform significantly better in low-density or small networks than AntHocNet.

A summary of the protocols' classifications is shown in Table 1.

3.3. Other nature-inspired routing approaches

Antoniou et al. [3] have proposed a flock-based congestion control algorithm for WSNs which is able to route data packets over a topological space whilst trying to avoid congested regions. The work models data packets as birds that create flocks and fly towards network destination (attractor sink node) following a direction of motion influenced by the repulsion and attraction forces exercised by the neighbouring packets, as well as the gravitational force in the direction of the destination. The flocking algorithm achieves low latency and fault tolerance while outperforming congestion-aware multi-path routing in terms of packet delivery ratio.

Particle Swarm Optimisation (PSO) [36] mimics flocking or schooling behaviour in nature and applies it to finding the shortest path in wireless networks. PSO is based on the idea that simple agent entities, termed particles, are randomly positioned in the parameter space of the problem. The position of each particle represents a candidate solution and by adjusting their trajectories, position and velocity, PSO can exploit the available population and probe the search space towards a desirable goal. Recent examples that apply PSO to wireless ad hoc routing are found in [46,39,56]. Similarly to PSO, Genetic Algorithms (GAs) [29] are population-based stochastic search engines that mimic natural selection and biological evolution processes. The current literature contains a number of works that prove the success of applying GAs in wireless sensor networks, in particular when energy-aware routing solutions are required [65,28,23].

Table 1
Summary of Swarm Intelligence based protocols in this paper.

Protocol	SI and network type	Forwarding	Scheduling	Reported strength
ARA	ACO, MANETs	Hop-by-hop	Reactive	Less overhead
Termite	ACO, MANETs	Hop-by-hop	Reactive	Load balancing
AndHocNet	ACO, MANETs	Hop-by-hop	Hybrid	PDR, delay
T-ANT	ACO, WSNs	Hop-by-hop	Reactive, cluster-based	Less memory, overhead
EEABR	ACO, WSNs	Hop-by-hop	Reactive	Energy efficient
HOPNET	ACO, MANETs	Hop-by-hop	Hybrid	Scalable
BeeAdHoc	BCO, MANETs	Source routing	Reactive	Energy efficient
BeeSensor	BCO, WSNs	Source routing	Reactive	Energy efficient
SLR	BCO, MANETs	Hop-by-Hop	Hybrid	PDR (up to 100 nodes)
BeeIP	BCO, MANETs	Source routing	Reactive	PDR, delay

Understanding the non-equilibrium nature of the self-organising wireless ad hoc networks, where nodes connect and disconnect to the network on a frequent basis, Palmieri and Castiglione [48] propose a novel approach for designing reactive routing protocols that maps Bose–Einstein condensation [10] to ad hoc network dynamics. The authors use AODV as a base, proposing a new routing strategy that is conditioned by the condensation phenomenon. Data packets are routed based on the nodes' capacity to acquire connections, and thus, are considered more likely to deliver packets to their destinations through reliable paths. The condensation-based version outperforms the conventional one in end-to-end delay, packet delivery ration and packets dropped, especially when the network load becomes higher.

4. The BeeIP routing protocol

BeeIP [26,25,27] is a new honeybee-inspired adaptive routing protocol based on the collaborative behaviours of honeybee foragers. Following a reactive approach, BeeIP's honeybee agents explore the topology only when data are required to be transmitted between nodes. They are designed to monitor and evaluate the performance of the discovered paths and select the optimal one based on a selection mechanism inspired by their natural counterparts.

Inspired by Swarm Intelligence, the proposed protocol makes a number of assumptions regarding the agents in use, their behaviour and finally the communication between them. These assumptions are designed to map the corresponding concepts of their natural counterparts. Apart from the obvious mapping; a source node is considered the honeybee hive, the destination is the flower or source of food, and the intermediate nodes the path or flying distance to be traversed from the hive to the flower, in BeeIP:

- the type of the missing commodity as it is monitored and signalled by the honeybees, i.e., nectar, pollen or water is mapped to the desired behaviour for selecting the optimal path based on the type of traffic for a communication session (e.g., the fastest path or the path with the least energy consumption),
- the foraging burden in the honey stomach is mapped to the data payload to be piggybacked to a forager packet,
- the level of satisfaction (the quality) of a food source and the foraging activity towards it is mapped to the quality of a path as calculated by the scout and the forager agents,
- the honeybee dance is mapped to the mechanism of making changes to the representation of how many data packets are allowed to use the particular path for the future transmissions (termed foraging capacity in BeeIP),
- the honeybee recruitment is mapped to the allocation of forager agents, to appropriate paths for future transmissions (according to the foraging capacity of each path).

BeeIP's novelty is found in the idea of constantly monitoring and evaluating the performance of each finding using previous experience by the scout and forager agents that explore the network, over time, that is, assessing the quality of the paths by comparing new quality measurements with previous ones. Several quality factors are considered by natural foragers. These factors are mapped to low-level parameters of the network, which are used to represent and measure the improvement or deterioration over time. Similar to nature, an improved path triggers more agents to work on it by using a metaphor of the honeybee recruitment.

Improvement over time offers an adaptivity which not only is a characteristic that can be mapped from nature to networks, it is also one of the fundamental ideas behind any adaptive routing protocol in the field of SI-based networking. Being able to respond to changes by adjusting routing, not only enhances the self-healing aspect of the protocol, it also aids its scalability to different network conditions and sizes.

BeeIP's agents are able to adaptively work in parallel, communicate with each other to exchange simple information that is carefully collected from the network and achieve multi-path routing at a global level. Loosing an agent (packet loss) or even an intermediate node of a path, is clearly a disruption to a proposed routing solution. Nevertheless, BeeIP's SI components allow the protocol to effectively tackle failures and invoke alternative solutions or initialize new scouting processes to find some new. Distribution is achieved and in turn, the computational costs are spread over the nodes that participate to each particular network activity.

The most important internal mechanisms of the protocol are discussed in the next section. Namely, the adaptive scouting, foraging, optimal path selection and detection of path failures.

4.1. Adaptive scouting

When a route is required at the source node and no previous information is available, a scout packet is generated and sent by broadcast to the rest of the network. The scout packet is responsible for discovering available paths to its destination, while it introduces neighbouring nodes to each other, in a hop by hop manner.

If the receiving node is the desired destination, an ack_scout packet is created and sent back to the source to confirm a successful path. Unicast is used instead of broadcast, because the ack_scout already knows which nodes to visit in order to return, that is, the reverse of the path the scout traversed in the first place. A destination node is allowed to create multiple ack_scouts, depending on the number of scouts it receives from the network, allowing multiple paths to be discovered.

Additionally, the destination node assigns to each discovered path a unique identification number. During the journey back, ack_scouts constantly collect and deliver low-level data received from the lower layers, from one node to the other.

Data transmission is achieved by creating, loading and sending forager packets. Each forager is responsible for carrying data payload received from the transport layer. At the source node, the selection mechanism for the most appropriate path depends on the desired behaviour of the protocol, subject to its implementation. The default selection is based on the fastest path between the source and the destination. At each intermediate node, the next hop is easily retrieved from the local routing information, based on the direction of the forager and the identification number that it carries. This approach is inspired by the source routing [24,53] met in other reactive protocols such as DSR [35]. However, in BeeIP the forager packet does not need to carry the whole path within its header. Instead, each intermediate node already knows which paths it is a member of, and based on the direction of the forager, it can decide the next node towards the source of the destination.

At the destination node, the forager packet becomes an ack_forager packet and stays in a buffer queue until it is again assigned to carry data back to the source. This behaviour mimics the real honeybees which stay on the flower for some time collecting the required commodity, such as pollen or nectar. For scheduling purposes they are utilised in a first-in-first-out fashion. The next hop selection for an ack_forager is then done again by considering the forager's direction.

Foraging in nature consists of several phases; a honeybee being recruited to follow a specific mission, flying to the target source of food, collecting the actual pollen, nectar or water and carrying it back to the hive. When a honeybee completes a foraging cycle, thus she is back at the hive, she can either perform a honeybee dance to share up-to-date information about the foraging, or forget it and focus her attention on something else. The decision is based on a number of the quality factors such as the sweetness of sugar, the purity of taste, the quantity and ease of obtaining, the distance, the type of the flower, etc. [69]. BeeIP foragers are designed to follow similar principles. In particular, ack_foragers are able to judge the quality of the path and detect any possible improvement or deterioration over time.

Monitoring a path's quality is both a complicated and network-dependent procedure, the outcome of which allows the source to adjust a path's foraging capacity. The foraging capacity is defined as the total number of foragers allowed to be recruited and use the path in the future. In more detail, every time a new artificial dance is released for a particular path, a small number of foragers is added (recruited) to the path's foraging capacity. This approach ensures that paths which release more positive dances will end up with higher foraging capacities and become more active in the long run, than others which release negative dances or no dances at all.

4.2. Adaptive foraging

The scouting process is considered successful when one or more ack_scout is received by the source node. BeeIP concentrates on monitoring and constantly evaluating multiple paths. When a forager is received by the destination node, it delivers piggybacked data to the transport layer and converts to an ack_forager. Like the real honeybees which take some time on the flower to collect the pollen or the nectar, the ack_forager stays at the destination node until some data packet needs to go back to the initial source. While travelling back home, it collects up-to-date information from the nodes it visits and the links between them. This allows it to monitor the overall path's quality and be able to report the finding back to the source.

The path is seen as a chain of links between the intermediate nodes, from the destination back to the source. While traversing it, the ack_forager is responsible for collecting data that represent not only the current state of its senders, but also, the network effectiveness of each intermediate link.

The list of the low-level parameters used by the protocol, followed by a brief explanation of their importance is given below.

1. The ack_forager's signal strength at the receiving node in dBms or Watts. Being a packet itself, when received it carries a signal strength. A weak signal strength is an indication of long distance between the nodes and/or intermediate obstacles that affect the transmission.
2. The moving speed of the sender (velocity) in m/s. A moving node can easily go outside the transmission range and cause weak or broken links. For the same reason, a node with a fixed position is more promising than one that moves around constantly and at high speed.
3. The sender's remaining energy level in Joules. Nodes with sufficient remaining energy are less vulnerable and better candidates for future packet transmissions.
4. The size of the MAC queue of the sender in bits. The precise size of the queue at the MAC layer of the sender is an indication of how busy the sender is in terms of traffic and network congestion.
5. The transmission delay between the sender and the receiver of a link in seconds. The use of time-stamps and synchronised clocks allows the measurement of the time an ack_forager requires to complete a transmission from the sender to the receiver of a link.

All these parameters can be extracted by accessing the appropriate layers of the OSI model using cross-layering techniques [34,50,17]. The application layer can provide both the moving speed and the remaining energy resource. The physical layer can provide the packet's signal strength upon receipt. Knowing the size of the MAC layer's queue and the allowed data rate as defined by the 802.11 network standards [11,8] of the wireless adaptor, the queueing delay can be found. Finally, the transmission delay is calculated by considering the time-stamps between sending and receiving the packet.

In order to be used effectively, the above parameters need to be normalised, which is done by applying linear transformation. It is important to notice that the moving speed and both queue size and transmission delay are adversely affecting the performance. A node's speed equal to zero does not affect the transmission, as it does not alter the distance between the source and the destination. This makes it preferable compared to a node of high moving speed. Likewise, a node which has the smallest MAC buffer size is considered free of congestion. Hence, inverse normalisation is used for these parameters. The role of each parameter is important but they differ in the way they affect the performance and thus the quality. This concept is also met in real honeybees' quality evaluation, where although the exact threshold values are unknown, it is empirically found that they play different roles and are combined together [69]. Therefore, to use them efficiently a way to express their relative importance needs to be found. In BeeIP's design, a weighting system using the machine learning technique, Artificial Neural Networks (ANN) [37] is used for this purpose. This was achieved by applying off-line supervised learning based on data sets produced by the wireless network. The goal of the training was to find a set of weights that cause the output from the neural network to match the actual target values of the data sets, as closely as possible. Full details of the weight value selection process can be found in [25].

The formula to calculate the quality q of a link from node j to k as traversed by agent b is shown in the following equation:

$$q_{jk} = sig'_b * w_{sig} + speed'_j * w_{speed} + energy'_j * w_{energy} + qd'_j * w_{qk} + txd'_{jk} * w_{txd} \tag{1}$$

where the prime numbers are the normalised values of the parameters (sig for signal's strength, etc.) and the $w's$ are the appropriate weights.

Following the discussion regarding the quality of a link, a new q is calculated at every node visited by the ack_forager, and when it finally arrives home, the quality of the path from the destination d to the source s can be expressed as:

$$Q_{ds} = \sum_{n=1}^{m-1} (q_{N_{n+1} \to N_n}), \quad [d = N_m,\ s = N_1] \tag{2}$$

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

The result obtained by Eq. (2) is a number that can be used to represent the current quality of the path, in terms of the five low-level parameters. However, it represents only one single flight. The nature of the network is such, that obtaining and using the results of a single packet transmission can be highly misleading. In BeeIP's architecture, results from a constant number of previous flights are collected, and based on them, the source is able to investigate whether there has been an improvement or deterioration to the path performance over time. Once there is sufficient amount of data available, the last step of the methodology is to apply regression analysis and in particular, Pearson's correlation coefficient [57]. The idea of using regression analysis is to catch any strong positive or negative correlation between the two variables, in this case: time and the quality of a path Q^{ds}. In such way any valuable changes can be statistically detected. Rewritten to fit BeeIP's design, the Pearson's r is calculated by:

$$r = \frac{\sum_{i=1}^{k}(t_i - \mu t)\left(Q_i^{ds} - \mu Q^{ds}\right)}{\sqrt{\sum_{i=1}^{k}(t_i - \mu t)^2}\sqrt{\sum_{i=1}^{k}\left(Q_i^{ds} - \mu Q^{ds}\right)^2}} \tag{3}$$

where t_i is the time of receiving Q_i^{ds}, μt the mean of the time column values and k the number of flights collected.

As real honeybees do in nature, their artificial counterparts must posses a mechanism to detect whether the path they are using remains worthy. In addition, following the principle of honeybee dancing, when they have enough evidence and a clear understanding of the path quality, they start the process of recruitment. In the protocol's context, this is translated to collecting enough data from previous flights. If the correlation is a strong negative, then the signal which is given to the recruits is also negative, reducing the path's foraging capacity. If the correlation is a strong positive, the foraging capacity is increased. The two thresholds for catching the strong correlations are set empirically to -0.8 and 0.8. Nonetheless, depending on the implementation these thresholds can be changed, altering the sensitivity of both the monitoring and foraging activities. Obviously, the closer to -1 and 1 these thresholds get, the less sensitive the protocol becomes.

4.3. Selection of optimal paths

Depending on the behaviour of the routing protocol that one may want to achieve, different selection metrics can be applied. On their way back, foragers collect that information and mark each path with a selection metric value. Traditional metric values are related to the number of hops in a path, the transmission speed of its links, the expected transmission count, the energy cost, the remaining energy, etc., [14,49,15]. Understanding the importance of this metric and its influence on the behaviour of the protocol is very significant. For the experimental comparison of this work, a metric related to speed is used; the summation of the (half-round) transmission delay and queueing delay for each intermediate link of the path, from the destination towards the source. This ensures that the fastest path from the list is selected.

This type of selection is done only at the source node. At the intermediate nodes, the selection is done by considering the direction and the path identification number. At the destination node the selection is done in a first-in-first-out fashion. This approach not only palliates the retransmission delays, but it also keeps the size of the packet small (compared to source routing), allowing other useful routing information to be accommodated, if necessary.

4.4. Detection of path failures

Despite the constant monitoring and evaluation of the paths, where agents can detect disturbances and prevent loss of connection by changing automatically to another alternative, links between the nodes can still break rather unexpectedly. An adaptive routing protocol ought to have a mechanism to detect when a link has been broken and update its routing knowledge.

Since BeeIP is designed to evaluate routing based on ''path'' level instead of ''link'' level, link breakage within a path is detected when no foragers return back to the source node within a period of time. The source node sets the path's foraging capacity to 0 and marks the path as unacknowledged. The first ensures that no future foragers will be given the broken path's ID, whereas the latter allows the path to become available again, if a forager eventually comes back. Furthermore, in order to get rid of very old unacknowledged paths, a timer is used for housekeeping. This simple mechanism ensures that control overhead remains low because no special messages need to be exchanged just to confirm nodes' existence. In terms of the intermediate and the destination nodes, a timer pruning is also triggered and unused routing information is removed.

5. Comparing BeeIP to BeeAdHoc: a qualitative discussion

The authors consider BeeAdHoc to be the most related work in the literature. Both BeeIP and BeeAdHoc are designed to work in wireless ad hoc networks of mobile nodes (MANETs) and are inspired by honeybees and traditional routing. Comparing the two together is expected to give extra merit to the strengths and features of BeeIP. However, BeeAdHoc source code is not available in the public domain for the network simulator that is used in this work, so a direct comparison cannot be performed. However, an attempt to qualitatively compare the two protocols is presented in this section. BeeAdHoc's published experimental results use traffic that is generated by constant bit rate over UDP and make use of the swarm packets only, without covering the TCP traffic and the use of piggybacked acknowledgement data packets to foragers. In [72], BeeAdHoc is compared to the state-of-the-art AODV, DSR and DSDV (no SI approaches are included, such as AntHocNet) using a similar configuration setup to the one used in this paper.

BeeIP's mechanism to measure the quality of a path considers five different factors, whereas in BeeAdHoc the quality measurement is based only on the delay between the links and the remaining energy at the nodes in a path. In addition, BeeIP keeps a recent history of previous quality findings. Combining those two features, BeeIP is able to detect improvements or deteriorations over time and make changes to the foraging capacity accordingly. BeeAdHoc does not consider previous knowledge nor detects improvement or deterioration on the path's quality. Rather, each forager may dance and recruit fellow foragers according to the new quality finding only. Mimicking nature, BeeIP is designed to apply penalties to the foraging capacity which permits the protocol to react faster to quality changes. Additionally, its flexible design in terms of next hop selection metric, makes BeeIP easier to implement and fit the needs of the different deployments. In an indirect comparison, the results of [73,72] show that BeeAdHoc is able to send less control overhead than the others, but achieve less or equal packet delivery ratio to DSR, whereas BeeIP is found to achieve better packet delivery ratio than DSR.

BeeAdHoc uses source routing which is reported as a known disadvantage of the protocol that increases the control packet size as the length of the route gradually increases [71]. BeeIP on the other hand uses a different approach where the next hop at the intermediate nodes is decided based on the path's unique identification number and the direction. Based on this logical argument, BeeAdHoc is considered less scalable in terms of number of nodes (longer paths in hops) as its foragers need to carry the whole path in their headers.

Another important design difference is that in BeeAdHoc each forager has a lifetime value. This value is considered when the protocol needs to send a matching forager to the packing floor in response to a request from a packer. The foragers whose lifetime has expired are not considered for matching [22]. Following this, in the fourth chapter of [71], Wang reports ''*Another real disadvantage [of BeeAdHoc] is the higher memory use for storing every forager. Although they are really small it is more than storing every route only once*''. Clearly, keeping foragers in memory is an expensive behaviour, especially when the devices in MANETs and WSNs come with limited resources. In BeeIP on the other hand, the foragers are destroyed when not used, deallocating their memory rather than keeping it; a source node keeps track of the number of foragers currently waiting (forgers in) and currently using the path (foragers out) for every stored entry of the routing repository. This design difference implies that BeeIP uses less memory than BeeAdHoc in terms of storing its control packets.

6. Comparing BeeIP to the state-of-the-art: a quantitative discussion

Testing the performance of BeeIP is done by thoroughly comparing it to four existing state-of-the-art routing protocols [25] which were introduced in Section 2. Namely, BeeIP is compared to the Ad hoc On-Demand Distance-Vector protocol (AODV), the Dynamic Source Routing (DSR), Destination Sequenced Distance-Vector (DSDV) and the multi-path Ad hoc On-Demand Multi-path Distance-Vector (AOMDV). Each protocol has its own characteristics and applies different routing strategies. In addition, the experiments are conducted using the ns-2 network simulator [47]. The methodology followed in this comparison is similar to the benchmark comparison proposed by Broch et al. [12], as well as several routing protocol comparisons found in the literature [6,4,55].

As a reactive protocol, BeeIP shares some common characteristics with AODV. In fact, the latter has been a source of inspiration in terms of the route request technique for finding new routes, as well as the gradually expanded broadcast of the scout packets, to manage the control overhead. Therefore, it has been decided that the results of their comparison will add valuable profit in understanding BeeIP. Since BeeIP is a multi-path protocol, the comparison to the multi-path approach, AOMDV, is expected to give extra value to the investigation. DSR has inspired BeeIP's packet forwarding approach, which divides the required next hop selection information to the intermediate nodes, reducing both computation and storage capacity. Additionally, DSR's draft has also an experimental status by the IETF MANET group and is used in the literature as a comparison counterpart for reactive routing protocols. Hence, the results of their comparison will enhance the understanding of BeeIP's performance. Comparing BeeIP to the proactive DSDV, not only complies with good common practice but also is expected to complete the picture of BeeIP performance differences under a variety of conditions.

In this paper, an experimental study that examines the adaptation to frequent link breaks and robustness of each protocol against the nodes' moving speeds is included. Table 2 summarizes the base ns-2 configuration and set-up for the comparison of the protocols. For 600 s, 100 nodes are moving according to the Random Waypoint model in a network area of $3000 \times 1000\ \text{m}^2$. The pause time is fixed to 75 s. In terms of traffic, 40 TCP sources are used to send data packets, making use of 60 connections. The experiments are repeated by gradually widening the speed range of the nodes, starting from 1–5 m/s, to 1–10 m/s, 1–15 m/s and 1–20 m/s. As the maximum speed that the nodes can reach is increased, the average speed is also increased, as well as the frequency by which links break. The reason behind this particular configuration is again to simulate a harsh condition, when some nodes will shut-down causing transmission links to break, and in turn lead to routing anomalies. Broken links cause extra delays and affect the overall throughput of the network. In addition, the recovery process after a link is found broken, requires extra control overhead to occupy the medium. Thus, the aim is to investigate how the protocols are able to adapt to the topological changes.

The performance metrics used for the comparison are:

1. *Packet delivery ratio*: This is the ratio between the successfully received data packets at the destination node, to the successfully sent data packets by the source node. This metric does not take into consideration the control packets and achieving a high result is required by any routing protocol.
2. *Control overhead*: This is the number of routing-related packets that are sent in order to exchange routing information and maintain routing. The less control overhead required by a protocol, the better.
3. *Average end-to-end Delay*: This is the difference between the time a packet is received by the destination node and the time it was originally sent by the source node. This includes all the intermediate delays a packet has to experience during the transmission, e.g., propagation delay, transmission delay, queuing delays, process delays, etc. A small average end-to-end delay means faster data transmission.
4. *Average throughput*: This is the average of the throughputs as calculated for each session between a TCP source and a TCP sink. The throughput is defined as the number of bits that have been delivered to a node for the duration of a session. The greater the average throughput, the better.

6.1. Results

Fig. 1 shows that BeeIP keeps a steady performance, slightly deteriorating as the mobility of the nodes increases. Its packet delivery ratio varies from $\sim 98.9\%$ to $\sim 99.6\%$ whereas the second best reactive protocol, AOMDV, fluctuates between $\sim 98.8\%$ and $\sim 99.4\%$. AODV and DSR are found to have more packet loss for highly dynamic networks, amongst the reactive protocols in comparison. Moreover, the proactive DSDV is also losing in delivering packets on time, as its packet delivery ratio dropped to $\sim 97.5\%$ for speed range 1–20 m/s.

In terms of control overhead in Fig. 2, DSR is found to be the least sensitive protocol. In addition, BeeIP, AODV and AOMDV increase the amount of packets required to maintain routing, with BeeIP achieving a lower overhead. The results obtained from this experiment show that the multi-path extension to AODV is more sensitive to high mobility networks, in terms of the number of control packets being produced. This agrees with the results obtained in [43], where the frequency of route discoveries and routing overhead is similar for AODV and AOMDV with varying mobility. Finally, increasing the average

Table 2
ns-2 simulation's set up.

Number of runs	10
Initial energy	200–1500 J
MAC layer	IEEE 802.11b DCF (CSMA/CA)
MAC interface queue size	50
PHY layer	914 MHz Lucent WaveLAN
Signal propagation model	Two-way ground reflection[a]

[a] Please refer to [62] for more information.

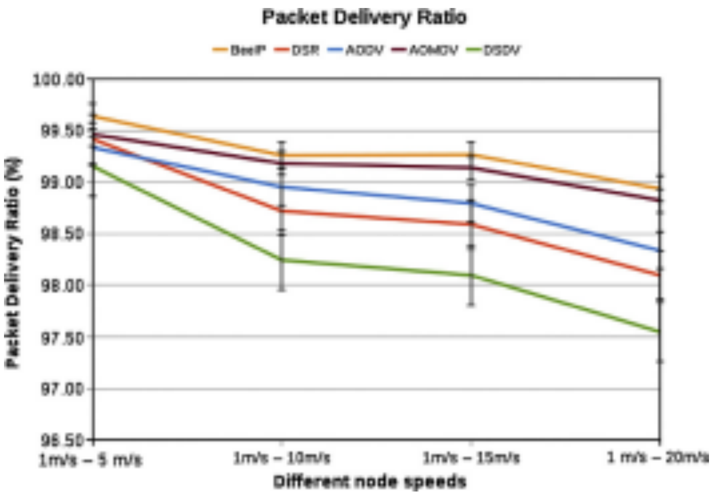

Fig. 1. Packet delivery ratio as a function of different node speeds.

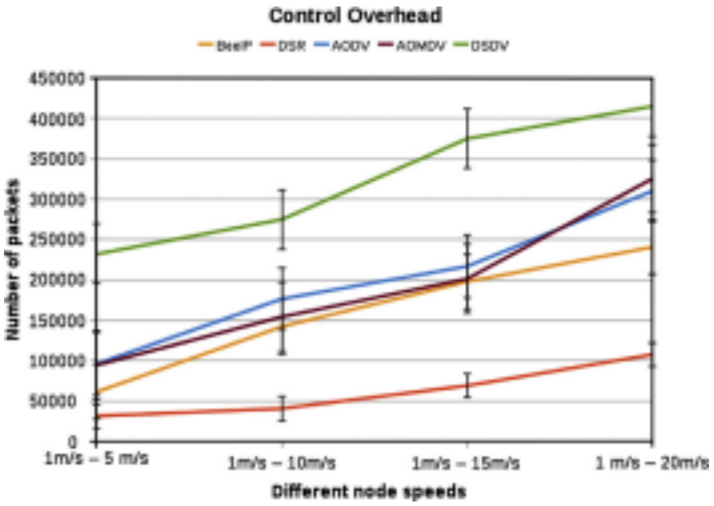

Fig. 2. Control overhead as a function of different node speeds.

speed of the nodes affects the proactive DSDV more, as route update messages flood the network in an effort to maintain routing.

In terms of end-to-end delay, the results are summarised in Fig. 3 and are as expected. DSDV follows the opposite trend compared to the reactive protocols, as its ability to adapt to consistently dynamic networks is quite limited. Again, BeeIP is

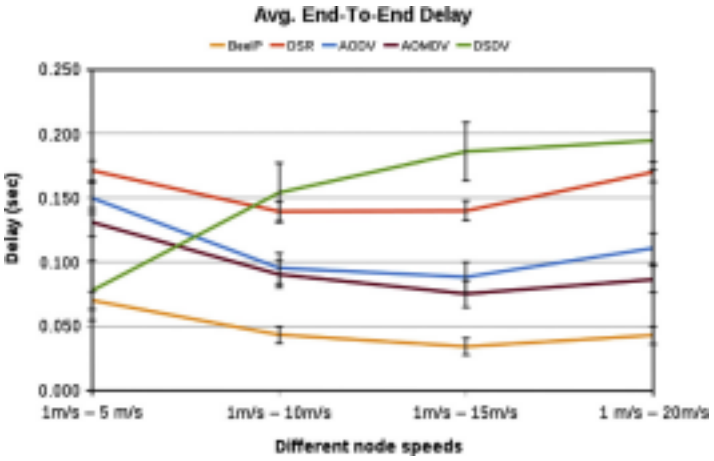

Fig. 3. Average end-to-end delay as a function of different node speeds.

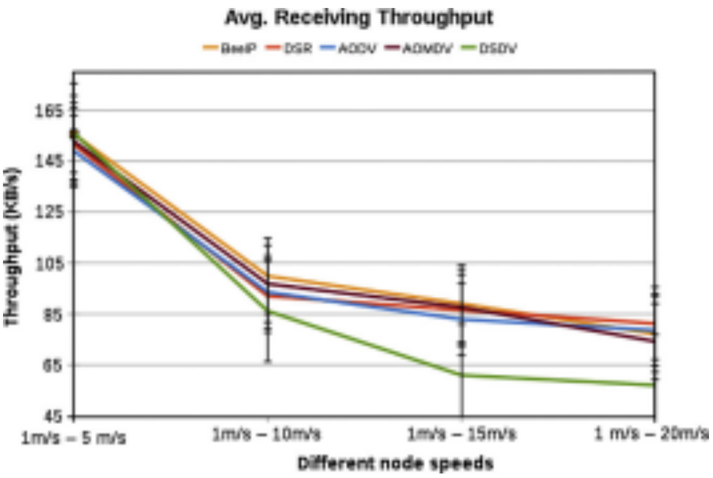

Fig. 4. Average throughput as a function of different node speeds.

shown to be able to perform better than AODV, AOMDV and DSR, achieving a balanced packet delay. Keeping in mind that the path selection metric of BeeIP is set to the fastest path, the low and balanced behaviour of BeeIP in connection to its high packet delivery ratio can be considered an indication of achieving efficient and optimal routing.

Considering the average throughput in Fig. 4, it is understood that the performance of all the protocols deteriorates as the average speed becomes higher. Due to the highly dynamic network when the node speed range is 1–20 m/s, BeeIP's average throughput is less than DSR and AODV. Nevertheless, under less stressful conditions, namely when the speed ranges are 1–5 m/s to 1–15 m/s, BeeIP is found to strongly compete with the other reactive protocols, achieving better performance ($\sim$ 155 KB/s, $\sim$ 100 KB/s, $\sim$ 89 KB/s), followed by AOMDV ($\sim$ 152 KB/s, $\sim$ 96 KB/s, $\sim$ 87 KB/s). To conclude, DSDV throughput is low, demonstrating its need to incrementally exchange route update messages to maintain routing, wasting a significant amount of bandwidth.

7. Conclusion

In this paper, several research milestones in the area of Swarm Intelligence based protocols for routing in wireless ad hoc and sensor networks are discussed. Inspired by Ant Colony Optimisation and Bee Colony Optimisation, the protocols are further classified based on their scheduling mechanisms and the network type in which they are designed to operate.

Furthermore, a new routing protocol inspired by honeybees is presented. The work emphasises the ability of honeybees to perform foraging and to communicate with each other within the hive, in order to achieve efficient and productive recruitment. It is an extended mapping of concepts and principle behaviours between nature and networks which allows routing between mobile wireless nodes in ad hoc telecommunication scenarios.

BeeIP is able to reactively discover multiple paths between sources and destinations, and distribute traffic across them in a scalable, robust and efficient way. The novelty of the system is seen by the way agents, which emulate the real scouts and foragers, constantly monitor and evaluate the performance of the previously discovered paths, over time. By the means of an artificial honeybee dance, and the use of statistical tools, the artificial honeybees are able to perform recruitment based on the path quality feedback as well as their past knowledge. In addition, deciding which is the most appropriate path to follow not only depends on what options are available from the dancing honeybees, but also on the particular need for a specific commodity (nectar, pollen or water). BeeIP has a flexible way of utilising a selection metric for the next hop at the source nodes, which is related to the particular behaviour the protocol is required to achieve. The latter is designed to be defined by the implementation of the protocol and depends on the application needs.

Adaptation is achieved as the protocol is able to make different routing decisions every time routing is required within highly dynamic networks. The nodes follow a decentralized and distributed approach in order to discover paths and forward packets, and routing becomes self-organised as no prior planning is required.

BeeIP was quantitatively compared to state-of-the-art routing approaches. The results obtained show that BeeIP generally outperforms the other protocols. Its biggest strength is seen when observing the average end-to-end delay and packet delivery ratio. The protocol is able to deliver packets faster, due to its packet switching mechanism and ability to discover and use multiple paths. These two characteristics allow the protocol to outperform the others within highly dynamic networks with increased node moving speeds. Additionally, the protocol maintains a balanced control overhead during mobility changes.

Acknowledgement

This research was funded by Aberystwyth University Postgraduate Research Fund.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Commun. Mag. 40 (8) (2002) 102–114.

[2] M. Amiri, Evaluation of Lifetime Bounds of Wireless Sensor Networks, 2010, Arxiv preprint arXiv:1011.2103 vol. abs/1011.2.

[3] P. Antoniou, A. Pitsillides, T.M. Blackwell, A. Engelbrecht, L. Michael, Congestion control in wireless sensor networks based on bird flocking behaviour, Comput. Netw. 57 (5) (2013) 1167–1191.

[4] B.R. ArunKumar, L.C. Reddy, P.S. Hiremath, Performance comparison of wireless mobile ad-hoc network routing protocols, Int. J. Comput. Sci. Netw. Secur. (JCSNS) 8 (6) (2008) 337–343.

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

[6] S.R. Biradar, K. Majumder, S.K. Sarkar, P. C, Performance evaluation and comparison of aodv and aomdv, Int. J. Comput. Sci. Eng. (IJCSE) 2 (2) (2010) 373–377.

[7] S. Blake, A. Pullin, The effect of node density on routing protocol performance in mobile ad-hoc networks, The Convergence of Telecommunications, Networking and Broadcasting, 2007.

[8] Board, IEEE-SA Standards. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band. Control 1999 (June), 2003. <http://standards.ieee.org/getieee802/download/802.11a-1999.pdf>.

[9] A. Boukerche, Algorithms and Protocols for Wireless and Mobile Ad hoc Networks, Wiley, 2009.

[10] C.C. Bradley, C.A. Sackett, J.J. Tollett, R.G. Hulet, Evidence of bose-einstein condensation in an atomic gas with attractive interactions, Phys. Rev. Lett. 75 (1995) 1687–1690.

[11] B.P. Brenner, A technical tutorial on the ieee 802.11 protocol director of engineering, Portal (January), 1997, pp. 147–152.

[12] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '98, ACM, 1998, pp. 85–97.

[13] T. Camilo, C. Carreto, J.S. Silva, F. Boavida, An energy-efficient ant-based routing algorithm for wireless sensor networks, in: Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence. ANTS'06, Springer-Verlag, 2006, pp. 49–59.

[14] M.E.M. Campista, D.G. Passos, P.M. Esposito, I.M. Moraes, C.V.N. de Albuquerque, D.C.M. Saade, M.G. Rubinstein, L.H.M.K. Costa, O.C.M.B. Duarte, Routing metrics and protocols for wireless mesh networks, Comput. Commun. 22 (1) (2008) 6–12.

[15] L. Cao, K. Sharif, Y. Wang, T. Dahlberg. Adaptive multiple metrics routing protocols for heterogeneous multi-hop wireless networks, in: Proc. IEEE CCNC 5th Consumer Communications and Networking Conference (IEEE CCNC'08), Las Vegas, NV, January 2008, pp. 13–17.

[16] G.D. Caro, M. Dorigo, AntNet: distributed stigmergetic control for communications networks, J. Artif. Int. Res. (JAIR) 9 (1998) 317–365.

[17] L. Chen, S.H. Low, M. Chiang, J.C. Doyle, Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks, IEEE, 2006.

[18] G. Di Caro, F. Ducatella, 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.

[19] M. Dorigo, T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, 2004.

[20] O. Dousse, P. Thiran, M. Hasler, Connectivity in ad-hoc and hybrid networks, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, vol. 2, IEEE, 2002, pp. 1079–1088.

[21] F. Ducatelle, Adaptive routing in ad hoc wireless multi-hop networks. Ph.D. thesis, Università della Svizzera italiana Faculty of Informatics2007.

[22] M. Farooq, Bee-Inspired Protocol Engineering, Springer, 2009.

[23] K.P. Ferentinos, T.A. Tsiligiridis, Adaptive design optimization of wireless sensor networks using genetic algorithms, Comput. Netw. 51 (4) (2007) 1031–1051.

[24] B.A. Forouzan, TCP/IP Protocol Suite, 3rd ed., McGraw-Hill, 2005.

[25] A. Giagkos, Protocol Design and Implementation for Bee-inspired Routing in Mobile Ad hoc Networks, Ph.D. thesis, Aberystwyth University, 2012.

[26] A. Giagkos, M.S. Wilson, BeeIP: Bee-inspired protocol for routing in mobile ad-hoc networks, in: S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, J.-B. Mouret (Eds.), From Animals to Animats 11, Proceedings of 11th International Conference on Simulation of Adaptive Behavior, SAB 2010, Paris - Clos Lucé, France, August 25-28, 2010, vol. 6226/2010 of LNAI, Springer-Verlag, 2010, p. 263–272. ISBN 978-3-642-15192-7.

[27] A. Giagkos, M.S. Wilson, Swarm intelligence to wireless ad hoc networks: adaptive honeybee foraging during communication sessions, Adapt. Behav. 21 (6) (2013) 501–515.

[28] V. Godbole, Genetic algorithm-based energy-efficient protocol for wireless sensor networks, Int. J. Ultra Wideband Commun. Syst.: IJUWBCS 2 (3) (2012) 133–140.

[29] D.E. Goldberg, Genetic algorithms in search, Optimization and Machine Learning, Addison Wesley, 1989.

[30] R. Groß, S. Nouyan, M. Bonani, F. Mondada, M. Dorigo, Division of labour in self-organised groups, in: Proc. of the 10th Int. Conf. on Simulation of Adaptive Behavior, Springer-Verlag, 2008, pp. 426–436.

[31] M. Gunes, U. Sorges, I. Bouazizi, ARA – The Ant-Colony Based Routing Algorithms for MANETs. IWAHN, 2002.

[32] Y.K. Hassan, M.H.A. El-Aziz, A.S.A. El-Radi, Performance evaluation of mobility speed over manet routing protocols, Int. J. Netw. Secur. 11 (3) (2010) 128–138.

[33] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wireless Commun. 1 (4) (2002) 660–670.

[34] M. Ibnkahla (Ed.), Adaptation and Cross Layer Design in Wireless Networks, CRC Press, 2009.

[35] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad-hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer, 1996, pp. 153–181.

[36] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, California, 2001.

[37] C.G.Y. Lau, Neural networks: theoretical foundations and analysis, IEEE Press, 1992.

[38] N. Lemmens, K. Tuyls, Stigmergic landmark routing: a routing algorithm for wireless mobile ad-hoc networks, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010), 2010, pp. 47–54.

[39] X. Li, W. Gang, L. Zongqi, Z. Yanyan, An energy-efficient routing protocol based on particle swarm clustering algorithm and inter-cluster routing algorithm for WSN, in: Control and Decision Conference (CCDC), 2013, pp. 4029–4033.

[40] C. Lie, J. Kaiser, A Survey of Mobile Ad hoc Network Routing Protocols, Tech. Rep. Series Nr. 2003-08, University of Ulm, 2005.

[41] P. Lucic, D. Teodorović, Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence, in: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, 2001, pp. 441–445.

[42] M. Marina, S. Das, On-demand multipath distance vector routing in ad hoc networks, in: Ninth International Conference on Network Protocols, November 2001, pp. 14–23.

[43] M.K. Marina, S.R. Das, Ad hoc on-demand multipath distance vector routing, Wirel. Commun. Mobile Comput. 6 (7) (2006) 969–988.

[44] S. Marwaha, C.K. Tham, D. Srinivasan, Mobile Agents based Routing Protocol for Mobile Ad Hoc Networks, IEEE, 2002.

[45] C.S.R. Murthy, B.S. Manoj, Ad Hoc Wireless Networks Architectures and Protocols, Prentice Hall, 2004.

[46] A.S. Nasab, V. Derhami, L.M. Khanli, A.M.Z. Bidoki, Energy-aware multicast routing in manet based on particle swarm optimization, Procedia Technol. 1 (0) (2012) 434–438. First World Conference on Innovation and Computer Sciences (INSODE 2011)..

[47] ns-2, The Network Simulator – ns-2, 2012, <http://www.isi.edu/nsnam/ns/> (accessed 16.04.12).

[48] F. Palmieri, A. Castiglione, Condensation-based routing in mobile ad-hoc networks, Mobile Inform. Syst. 8 (3) (2012) 199–211.

[49] A.P. Patil, K.R. Kanth, B. Sharanya, M.P.D. Kumar, J. Malavika, Design of an energy efficient routing protocol for MANETs, IJCSI Int. J. Comput. Sci. Issues 8 (4) (2011) 215–220.

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

[51] C.E. Perins, P. Bhagwat, Highly Dynamic Destination-sequenced Distance Vector (DSDV) for Mobile Computers, ACM SIGCOMM '94, 1994.
[52] C.E. Perkins, E.M. Royer, S.R. Das, Ad hoc on demand distance vector routing. RFC 3561, 1999. <http://www.ietf.org/rfc/rfc3561.txt>.
[53] J. Postel, Internet Protocol – DARPA Inernet Programm, Protocol Specification. RFC 791, September 1981, <http://www.ietf.org/rfc/rfc791.txt>.
[54] S. Radha, S. Shanmugavel, Effect of exponential and normal distributed speed and position for mobiles in ad hoc networks, Model. Des. Wirel. Netw. 4531 (1) (2001) 293–300.
[55] A.H.A. Rahman, Z.A. Zukarnain, Performance comparison of AODV, DSDV and I-DSDV routing protocols in mobile ad hoc networks, Eur. J. Sci. Res. 31 (4) (2009) 566–576.
[56] C. Rajan, N. Shanthi, Swarm optimized multicasting for wireless network, Life Sci. J. 10 (4s) (2013).
[57] T.R. Read, N.A. Cressie, Goodness-of-fit statistics for discrete multivariate data, Ser. Stat., Springer, 1988.
[58] M. Roth, S. Wicker, Termite: Ad-Hoc Networking with Stigmergy, IEEE, 2003.
[59] M. Roth, S. Wicker, Termite: A Swarm Intelligent Routing Algorithm for Mobile Wireless Ad-Hoc Networks, Cornell University, 2005.
[60] M. Saleem, M. Farooq, BeeSensor: A Bee-Inspired Power Aware Routing Protocol for Wireless Sensor Networks, Springer-Verlag, 2007.
[61] M. Saleem, M. Farooq, A framework for empirical evaluation of nature inspired routing protocols for wireless sensor networks, in: IEEE Congress on Evolutionary Computation, 2007. CEC, September 2007, pp. 751–758.
[62] T.K. Sarkar, A. Medouri, M. Salazar-Palma, A survey of various propagation models for mobile communication, IEEE Antenn. Propag. Mag. 45 (3) (2003) 51–82.
[63] S. Selvakennedy, S. Sinnappan, The time-controlled clustering algorithm for optimized data dissemination in wireless sensor network, in: The IEEE Conference on Local Computer Networks, 2005, 30th Anniversary, November 2005, pp. 2–510.
[64] S. Selvakennedy, S. Sinnappan, Y. Shang, T-ANT: A Nature-Inspired Data Gathering Protocol for Wireless Sensor Networks, IEEE, 2006.
[65] N. Sudha, M.L. Valarmathi, T.C. Neyandar, Optimizing energy in wsn using evolutionary algorithm, in: Proceeding of IJCA on International Conference on VLSI, Communications and Instrumentation, Kerala, India, April 2011, pp. 26–29.
[66] N. Tantubay, D.R. Gautam, M.K. Dhariwal, A review of power conservation in wireless mobile adhoc network (MANET), IJCSI 8 (4(1)) (2011).
[67] D. Teodorović, T. Davidović, M. Selmić, Bee colony optimization: The applications survey, ACM Transactions on Computational Logic, 2011, pp. 1–20.
[68] C.N. Ververidis, G.C. Polyzos, Impact of node mobility and network density on service availability in manets, in: Proc. 5th IST Mobile Summit, June 2005.
[69] K. von Frisch, The Dance Language and Orientation of Bees, Oxford University Press, 1967.
[70] J. Wang, E. Osagie, P. Thulasiraman, R.K. Thulasiram, Hopnet: a hybrid ant colony optimization routing algorithm for mobile ad hoc network, Ad Hoc Netw. 7 (4) (2009) 690–705.
[71] H.F. Wedde, M. Farooq, J. Fischer, M. Kowalski, M. Langhans, N. Range, C. Schletter, R. Tarak, M. Tchatcheu, C. Timm, F. Volmering, S. Werner, K. Wang, B. Zhai, BeeAdHoc efficient/secure/scalable routing framework für AdHoc Netze, Tech. Rep. PG460, LSIII, School of Computer Science, University of Dortmund, 2005.
[72] H.F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, R. Jeruschkat, Beeadhoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior, in: GECCO, 2005, pp. 153–160.
[73] H.F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, R. Jeruschkat, BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior, in: Proceedings of the 2005 Conference on Genetic and Evolutionary computation, GECCO '05, ACM, 2005, pp. 153–160.
[74] N. Xu, A survey of sensor network applications, IEEE Commun. Mag. 40 (2002).