BeeWS: honeybee-inspired, large-scale routing protocol for wireless sensor networks (WSNs)

Fatih Çelik* and Ahmet Zengin

Department of Computer Engineering,
Sakarya University,
Campus 54187 Serdivan/Sakarya, Turkey
Email: fatihc@sakarya.edu.tr
Email: azengin@sakarya.edu.tr
*Corresponding author

Abstract: Recently, scalable routing protocols using swarm intelligence (SI), which are designed and evolved for wireless sensor networks (WSNs), have become a research trend in literature. This paper presents an optimal method based on SI inspired by honeybees. This method has minimum energy consumption, self-organisation, and support for a large-scale, autonomous individuals that detect the best route. In this paper, we propose a new energy-aware, scalable, and robust routing algorithm called Bee wireless sensor (BeeWS) inspired by honeybee foraging behaviour. This study consists of three parts: (1) honeybee behaviours are modelled and these behaviours are adapted to the structure of WSN; (2) routing protocol criteria are determined using this model and (3) the developed model is tested in a simulation environment based on discrete event system specification (DEVS) in order to simulate and model WSN behaviours and compare to other SI based on WSN routing protocols.

Keywords: SI; swarm intelligence; routing protocol; WSNs; wireless sensor networks; clustering.


Biographical notes: Fatih Çelik received his PhD in Electronic and Computer Education at the University of Sakarya. He is an Assistant Professor at the Department of Computer Engineering, University of Sakarya, Turkey. His research topics include DEVS theory, multi-formalism modelling, parallel and distributed simulation, modelling and simulation of large-scale networks, distributed systems management, biologically-inspired optimisation schemes and wireless sensor network. His main research interest lies in parallel and distributed simulation and routing protocols for wireless sensor network.

Ahmet Zengin received her PhD in Electric and Electronic Engineering at the University of Sakarya. He is an Associate Professor at the Department of Computer Engineering, University of Sakarya, Turkey. His experience with modelling and simulation includes a one-year-stay in ACIMS Lab at the Arizona State University. His research topics include DEVS theory, multi-formalism modelling, parallel and distributed simulation, modelling and simulation of large-scale networks, distributed systems management, biologically-inspired optimisation schemes. His main research interest lies in parallel and distributed simulation and the high level architecture.

1 Introduction

Recently, technology of sensor nodes that comprise communication abilities, small size properties, and sensing precision have been followed up with developments in the micro-electronic industry. When these devices have been deployed in the environment, they have the potential for the ability of self-organisation and establishing communication with each other in sensing environments. Therefore, they have been used in military, environmental, health, and commercial applications. These applications often aim to monitor an environment (Çelik et al., 2010).

Routing in wireless sensor networks (WSNs) is one of the most important tasks due to limited processing, limited energy
supply, and limited communication in sensor nodes (Çelik et al., 2010). Route processing must carry out a model that reaches much more target data using the least communication (Al-Karaki and Kamal, 2004). In highly distributed and parallel environments, such as WSN systems, management must be accomplished without any central authority on distributed nodes (Akkaya and Younis, 2005). At the same time, it must prolong the network lifetime and respond to increasing scale (Saleem et al., 2011).

Insect colonies “consist of minimalist, autonomous individuals that through local interactions self-organise to produce system-level behaviours that show life-long adaptivity to changes and perturbations in the external environment. Moreover, these systems are usually resilient to minor internal failures and losses of units, and scale quite well by virtue of their modular and fully distributed design. All these characteristics, both in terms of system organisation and resulting properties, meet most of the necessary and desired properties of routing protocols for next-generation networks” (Farooq and Caro, 2008).

The contribution of this paper is a new large-scale BeeWS routing protocol, based on swarm intelligence (SI) inspired by honeybees for WSNs. A model inspired by honeybees’ colonies aims to discover the best routes using agents (Zungeru et al., 2012). While routes are discovered by a routing protocol, network life is affected little by agents, and agents reach the shortest path to the target. When the number of nodes increases in WSN, developed clustering protocols divide WSNs autonomous systems and find the best routes among cluster heads using BeeWS protocol. The BeeWS protocol can self-organise, adapt the WSN according to network changes, and divide to WSN autonomous systems.

The BeeWS protocol is designed and developed in the following three levels:

- first, honeybee behaviours are modelled and adapted to a WSN
- routing protocol criteria are determined using the adapted model, allowing the routing protocol to be developed according to these criteria
- the BeeWS routing algorithm is tested by a simulation framework based on DEVS (Zeigler et al., 2000) that is developed to simulate and model WSN behaviours using real world data.

Also, the results are compared to the well-known routing protocols FP-Ant (Zhang et al., 2004), EEABR (Camilo et al., 2006), BeeSensor (Saleem et al., 2012) and AODV (Perkins and Royer, 2000) that were developed for an ad-hoc network. It has been shown that the BeeWS protocol is better than these well-known protocols.

This paper is organised as follows. In Section 2, we briefly summarise the well-known routing protocols based on SI and clustering protocols for WSN. In Section 3, we correlate between sensor nodes and honeybees. Honeybees’ behaviours are modelled over sensor nodes. In Section 4, the BeeWS routing protocol architecture and functions are presented. In Section 5, the simulation environment is presented. The BeeWS is compared to well-known protocols, and the results are evaluated and discussed. In Section 6, we conclude with the paper’s contribution and a preview of our future research.

2 Related work

We now briefly summarise well-known routing protocols of SI and clustering for WSN. For brevity, we discuss only a few protocols, but an interested reader is referred to relevant survey papers and the references therein.

2.1 SI protocols for WSNs

The energy-efficient ant-based routing algorithm (EEABR) has been developed by Camilo et al. (2006). The main issue was to solve energy consumption. In addition, it was to solve the issues of network life and fault tolerances. It uses multipath to reach the destination. Sensor-driven and cost-aware ant routing (SC-ANT) aims to predict the best route with the ants moving in the forward direction on this route. Also, it strengthens the possible distribution (Zhang et al., 2004). The SC-ANT approach has energy consumption as a criterion. Flooded forward ant routing (FF-ANT) searches for a route using control packets to find the target and reach all nodes in the network. If the target is unknown or cannot be estimated, ants must roam the entire area (Zhang et al., 2004). The FF-ANT approach has delay as a criterion. It was discovered by evolving the SC-ANT approach. Flooded piggybacked ant routing (FP-ANT) lists in the name of the ants carrying the data revealed a new species of ant. The data ants routing process is similar to FF-ANT (Zhang et al., 2004).

BeeSensor, developed by Farooq and Saleem, inspired the Beehive routing protocol (Karaboga and Akay, 2009), which was developed for wired networks (Saleem et al., 2012). It is the first routing protocol based on SI for WSNs. The main goal was to solve energy consumption and scalability issues. BeeSensor uses four different agents. These are called packers, scouts, foragers, and swarms (Saleem et al., 2012). Packers work as static agents because they are placed in the node. Scouts explore to find routes. Scouts consist of forward scouts and backward scouts. Foragers use path information to carry packages, which are prepared by the packers. Swarms are similar to foragers. If foragers have more loads, swarms can carry the loads, as they can carry more loads than foragers.

2.2 Clustering based on routing protocols for WSNs

Low-energy adaptive clustering hierarchy (LEACH) is the best-known clustering method for WSNs (Heinzelman et al., 2002). Cluster heads route nodes toward the base station, and clusters are created according to the level of the signal strength. The LEACH is a distributed algorithm where nodes are autonomous and without a centre. Cluster heads are considered to reach the base station in a single hop. Therefore, it is not applied to a real environment. Energy-efficient cluster formation protocol (EECF) (Chamam and Pierre, 2009) and energy-efficient strong head clustering (EESH) (Wei
et al., 2007) using the method of multi-hop is clustering protocols. Energy efficient hierarchical clustering (EEHC) has been developed by Maria (1997). The EEHC method aims to increase the network lifetime by a random clustering algorithm. Cluster heads communicate to the base station by reporting information that receives member nodes. It has two type methods, input and extension. In the input method, some nodes are selected as cluster heads. They are called volunteer cluster heads. Cluster heads broadcast other nodes. Member nodes are connected to the nearest cluster head. Cluster heads that are not connected to anything are called strong cluster heads. The extension method creates hierarchy clustering.

3 From honeybee colonies to WSNs

In this paper, SI approaches are derived from honeybees. These interactions were developed by considering the mode of operation used by scouts that look for nectar fields in their foraging activities (Seely, 1995). The idea is based on “Real bees can find optimum solutions in their foraging activity” (Seely, 1995).

The behaviour of nectar collecting in honeybees’ colonies is a good example to investigate behaviours of social insects, such as self-organisation (Seely, 1995). Worker bees that consist of a section of the honeybee colony are employed to reach pollen and nectar sources. Worker bees are given a duty with a self-organisation method. Selection and separation between honeybees in the hive is done in the absence of any central governmental authority. Behaviours of honeybees can be adapted through simple rules based on criteria, such as the distance and duration of the exploration as well as nectar concentration, decentralisation and synchronicity. Parameters, including criteria, such as the number of honeybees, which is responsible for nectar storage in the hive, determine the profitability of the nectar source. If the colony encounters more than one nectar source, foragers can prefer the most profitable nectar source. Foragers, which use the criterion of profitability during the process of collecting nectar, are headed toward nectar sources. If nectar quantity varies by a certain rate, the importance of the nectar source for all colonies is reduced. In addition, nectar-searching honeybees are called as scouts. They are a small part of the colony. Scouts monitor the nectar fields and observe rich sources in the environment (Anderson, 2001). The assignment of the forager bees for food sources according to the profitability criterion is known as the scout-recruit mechanism in honeybees. One of the most well-known mathematical models of the honeybee’s scout-recruit system was developed by Seely (1995).

The basic activities of honeybees consist of exploration, operation, and travelling to and from nectar resources. The behaviour of honey bees consists of particular stages: travelling from the hive to the food sources, observation for nectar, nectar collection from the source, returning to the hive, transmitting environmental conditions to other bees, adapting to changes in the hive, and responding to environmental changes.

There is a similarity between WSN and honeybees as shown in Table 1. The WSN corresponds to honeybee colonies, which explore the nectar sources to reach rich food sources (Saleem et al., 2012). Each node in the WSN is similar to a hive. Transmission areas correspond to nectar areas of the honeybees. Sensor nodes in the WSN exchange data and control packets among themselves. Each node sends scouts to find the best route. Data packets correspond to foragers that carry nectar to the hive. Data packets are sent from the source to the destination using routing information, which is generated by control packets. Honeybee colonies are adapted to the WSN as shown in Figure 1.

Figure 1 The network structure of a hive (see online version for colours)
Table 1: Similarity between WSN and honey bees colonies

<table>
<thead>
<tr>
<th>WSN</th>
<th>Honeybee colonies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>Beehive</td>
</tr>
<tr>
<td>Transmission area</td>
<td>Fly to nectar/hive</td>
</tr>
<tr>
<td>Control packets</td>
<td>Scouts</td>
</tr>
<tr>
<td>Data packets</td>
<td>Data generator, data transporter</td>
</tr>
<tr>
<td>Routing tables</td>
<td>Dances and cues</td>
</tr>
</tbody>
</table>

4 Protocol description

Three different types of honeybee agents roam in the WSN. These are the data generator, scouts, and data transporter.

Data generators load data packets, which are generated using the path information from the data transporter. They send data packets from the routing layer to the transport layer. They complete tasks after delivering data packets to the data transporter.

Scouts are the control packets. They are very important to ensure survivability in the WSN. This situation is similar to the nectar search, which is very important to the survival of the honeybee colonies. In addition, control packets stabilise the network load.

Self-organisation of artificial honeybees has simple rules that are derived from the behaviour of the individual insects. It corresponds to the mechanical behaviour of a custom class, which is called the honeybee scout-recruit system. It is a response to the received information from the environment in accordance with predetermined rules. The BeeWS algorithm has the following rules:

- **Rule 1**: Every hive sends scout bees to gather information regarding nectar sources and selects the best route as an event triggered (see Algorithm 1).
- **Rule 2**: The purpose of scout bees is nectar collection during the collection process.

---

**Algorithm 1**: Algorithm for launching scouts

**Input**: source_node_id, data_number, node_energy

**Output**: Create Forward Scout

1. Create new packet format called IIZC (forward scout)
2. packet_route ← source_node_id
3. packet_id ← data_number
4. packet_source_address ← source_node_id
5. packet_energy ← node_energy

**Algorithm 2**: Algorithm for processing scouts

**Input**: scout_id, node_id, data_number, energy, cost,

**Output**: Scout process

1. if scout is forward scout & forward scout source is this node & forward scout is not in list of scout id then
2. energy ← energy
3. scout_id_list ← scout_id_list
4. if forward scout is in a hop distance then
5. packet_route ← source_node_id
6. packet_id ← data_number
7. packet_generated_address ← node_id
8. packet_target_address ← target_node_id
9. packet_cost ← total_cost
10. packet_energy ← total_energy
11. if cost is bigger than 15 then
12. scout discard
13. else
14. cost ← cost + 1
15. IIZC is sended
16. if scout is backward scout & backward scout source is this node then
17. GIZC is sent as unicast
18. target_node update
Algorithm 3: Algorithm for updating routing table

Input: route, node, energy, hop_count,
Output: Update Routing Table

1 if scout is forward scout & forward scout source is this node & forward scout is not in list of scout id then
2     if forward scout is in a hop distance then
3         table_update ← route ← next_node ← previous_node ← energy ← hop_count + 1
4     else
5         table_update ← route ← next_node ← energy ← hop_count + 1
6 if scout is backward scout & backward scout source is this node then
7     if source_node then
8         route ← route_code ← total_energy ← hop_count
9     table_update ← route ← next_node ← energy ← hop_count
10 else
11     table_update ← route ← previous_node ← total_energy ← hop_count

- Rule 3: Scout bees gather information concerning nectar source and routes.
- Rule 4: The amount of nectar that is collected along a certain route has an inverse ratio to the cost of the route in the WSN.
- Rule 5: When a scout reaches sink, it returns to its source from the previous route (see Algorithm 2).
- Rule 6: The scout never visits a sensor node twice. Therefore, the list of visits of scout bees is stored by every sensor node (see Algorithm 2).
- Rule 7: If a scout bee reaches the maximum hop counts, it dies (see Algorithm 2).
- Rule 8: When all or some of the scouts return to the hive, they bring route information and route cost. This information is stored (see Algorithm 3).

The scout’s task is to find a new route to send data packets. The data generator checks whether there is sink in a hop distance when it creates a data packet. If the sink is a hop distance, the data packet is loaded to the data transporter. If the sink is not a hop distance, the scouts are activated. Scouts are broadcasted to neighbours. Scouts have two types, forward scouts and backward scouts. Forward scouts explore all routes to sink. The flooding technique is used in this process. Forward scouts, which find a hop distance to a sink node, are transformed into backward scouts and return to the previous route.

The backward scout conveys route information and the route’s total energy information. The timer is started by the backward scout, which first arrives to the source node and waits a certain period of time. Route and energy information, which were obtained, are calculated and enumerated. The most appropriate route is selected from the routes that were enumerated. Data packets are loaded to the data transporter and sent through the most appropriate route. Data transporters must choose the most appropriate route. For this reason, the network lifetime is important.

4.1 Routing model of node based on proposed BeeWS algorithm

The proposed BeeWS algorithm model has a routing layer model that comprises an input section, evaluation section, dance section, and packaging section (Figure 2). Data packets arrive from the data link layer to the input section. They are classified according to type. Necessary additions are added to the packets, and packets are sent to the packaging section. If the packages arriving at the packaging section are scouts, they are sent to the evaluation or dance section. Otherwise, they are sent to the transport layer. If they arrive at the source node, they can be sent to the dance section to store packets that contain route codes, total energy, and hop counts. Source node switches to standby status. Otherwise, the sensor node stores packets’ previous node, total energy, and hop counts. They are sent to the packaging section.

The input section receives packets from the media access control (MAC) layer. Packets are classified as data generator, scout, and data transporter. Classified packets are sent to appropriate sections according to the type. If scout arrives at a source node, it can switch to standby state. Otherwise, it is sent to the packaging section. If packet is a data transporter and the sensor node has route information, it is sent to the packaging section to send next sensor node. If the data transporter is in a hop distance, it is sent to the packaging section without route checking. If the data transporter has any of these conditions, it is sent to the packaging section to activate scouts.

Data packets that arrive at the sensor node evaluate the evaluation section. If the packet is a data packet and there is a sink node in the sensor node’s transmission range, the packet is loaded a data transporter. Otherwise, routes towards the sink sensor node are checked, because route information is stored for a while. If there is route information, the route information is used. Otherwise, the forward scouts are activated to find new route. If packet is forward scout, the node broadcasts and the package is stored. If the node is the last node before the sink, a backward scout is created. If the node makes unicast, the
package is stored, and the backward scout is sent back to the same node. If the packet is a backward scout, the node makes unicast the package is stored. The backward scout is sent to the registered route in the node.

Figure 2 Model of node based on proposed BeeWS algorithm (see online version for colours)

The dance section is created in the source node. If a backward scout arrives at the source node, it stores the route’s hop count and total energy in the source node. A route number is added after this information is evaluated. The timer is set after the backward scout arrives the first time and a period of time is expected to take place at an average of 15 hops. During this period, the backward scout that arrives at the source node is evaluated and enumerated. At the end of that time, the timer is reset. Forward and backward scouts are expired in the source node. The route that has the least cost is preferred. The packet is loaded the data transporter.

If the packet is loaded the data transporter, it is sent to the transport layer. Otherwise, it is sent to the transport layer as a forward or backward scout according to type of packet. The packaging section establishes the relationship with the transport layer. If the packet that arrives at the packaging section is the data transporter, the packet is sent to the transport layer. If the packet is a scout, the type of the node is checked. If the node is a source node, the packet is sent to the dance section. Otherwise, the packet is sent to the evaluation section.

4.2 Clustering process

Nodes close to sink consume more energy. If sink were centre of deployment area, total energy would be more in nodes close to sink for more nodes will be around the sink. Total energy is less for nodes close to sink is less in hard-to-reach. This case affect the performance of the network lifetime (Marks, 2010). Therefore, clustering method was developed for prolonged network lifetime. The total energy saving was achieved by changing the cluster head closest to sink in certain phases using clustering method. So, use of only certain nodes close to sink was prevented. In addition, a large amount of energy loss occurs in cluster heads because the majority of the traffic on the clustering methods are guided by cluster heads. So, the number of clusters is important. It must be held for a suitable value of the number of clusters. This value is determined as 5% by LEACH.

New hives are constructed using the swarming method when the number of individuals in the hive increases. For that reason, the proliferation of the number of honeybees does not affect the operation of the system. In this study, the WSN was divided into clusters using the clustering algorithm.

Before the clustering process starts, all nodes broadcast HELLO packets to be aware of nodes in transmission range. The HELLO packets contain neighbour node ID, energy, and life expectancy. While HELLO packets wander around in the WSN, all nodes accept themselves as a cluster head. Thus, each node determines the nodes in the transmission range. If node has a number of neighbouring nodes, which is 15 and over, it broadcast itself as a cluster head, and nodes connect to cluster heads in transmission range. If a node is reached by two or more cluster heads, it connects to the cluster head that has the best connectivity rate. A cluster head has a transmission range of three times more than member nodes. Subsequent operations are updated at regular intervals by considering neighbours’ energy. When a cluster head reduces energy, the node having the maximum number of neighbours and the most energy is transferred to the cluster head task. In Figure 3, green rounds are member nodes. Yellow rounds are cluster head nodes (CH). Red rounds are sink nodes. Purple transparent rounds are cluster head transmission areas. Red transparent rounds are the member node transmission areas.

Figure 3 BeeWS cluster formation in DEVS-Sensor (Çelik, 2012) visualisation area (see online version for colours)

Data generators create data packets, and data transporters carry data packets to cluster heads after member nodes sense environment. Cluster heads create the WSN among themselves, and they reach sink node using a multi-hop method. Cluster heads in the WSN use BeeWS architecture during the operation period.

In order to spread the use of energy over the nodes in a balanced manner, cluster heads among same cluster nodes
were changed at certain intervals. So, a set $CH$ of nodes might elect themselves cluster heads at time $t$, but at time $t + d$ a new set $CH$ of nodes elect themselves as cluster heads, as shown in Figure 4. The decision to become a cluster head depends on the node having the maximum number of neighbours and the most energy remaining at the node. Hence, nodes remaining more energy will fulfill the intense payload of the network. Each node makes its decision about whether to be a cluster head independently of the other nodes in the network and thus no extra negotiation is required to determine the cluster heads.

Figure 4 Dynamic clusters: (a) cluster heads and member nodes at time $t$ in phase 1 and (b) new cluster heads and new member nodes at time $t + d$ in phase 2 (see online version for colours)

5 Simulation framework

The developed DEVS-Sensor simulation framework and BeeWS routing algorithm’s outstanding features are proven in this section, tests and experimental results regarding BeeWS non-clustering and clustering protocol are presented.

The BeeSensor algorithm is based on a bee colony algorithm among the developed algorithms for WSNs in literature (Çelik et al., 2010). The EEABR is an SI algorithm, which gives efficient results in the field of energy in literature. The FP-Ant is an algorithm based on SI, which gives efficient results in the field of packet delivery in literature. The AODV is an algorithm that makes up the basic principles in the field of wireless tools.

For the above reasons, the BeeWS that was designed as a swarm-based routing algorithm for WSNs was compared to the EEABR, FP-Ant, AODV, and BeeSensor algorithms according to the number of nodes in terms of energy, latency, and packet delivery. Also, protocols is compared by changing the density of packets. That’s why, we performed simulations in terms of latency and packet delivery.

Latency is defined as the time difference between an event that is generated in a source node and its delivery to the sink node. We do not include lost events in calculating the latency. The packet delivery ratio is the ratio of the total number of events received by a sink node to the total number of events generated by all source nodes in the network. Energy efficiency is the energy consumed by 1000 bits of data that is delivered to a sink node (J/Kbits) (Ammer and Rabaey, 2006). The results obtained are shown graphically.

Scaling is one of the most important points in the study. We are able to establish large-scale WSN and run the developed clustering algorithm in the simulation framework. The obtained results, including energy, latency, and packet delivery ratio, are graphically shown.

5.1 Simulation parameters

The BeeWS routing algorithm was simulated using the DEVS-Sensor simulation environment, which is developed using DEVS. The DEVS is discrete event modelling method that is the most appropriate to construct a WSN.

The BeeWS routing algorithm was compared to the EEABR, FP-Ant, AODV, and BeeSensor. Application scenario was used in a converge-cast scenario. We have evaluated five protocols in a typical converge-cast scenario in which multiple sources communicate with a single global sink. We preferred a scenario that is hard-to-reach areas. Sink is located in outside of hard-to-reach areas for the data received from the nodes send to user via internet or the other way. If a sink is located in the centre of WSN, the data gathered cannot be sent to user in hard-to-reach areas. So, sink was located in the edges of the area (see Figure 3) (Al-Karaki and Kamal, 2004). The event in the application is 512-bits long and generated at a rate of one event per second (Chen and Khokhar, 2004).

We report the results of our simulation studies in two phases. In the first phase, we reported the empirical results after the BeeWS is compared to the EEABR, FP-Ant, AODV, and BeeSensor. In the next phase, we evaluated the BeeWS clustering in bigger topologies and compared its performance with BeeWS non-clustering.
In each of these experiments, we assumed symmetric links (Chan et al., 2004) in the network where nodes were deployed randomly. Each experiment was performed over the duration of 300 s, and the reported values were an average of 10 independent runs. Berkeley mote CC2420 MICAz Wireless Sensor parameters were used in these experiments (Table 2).

Table 2 MICAz parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MICAz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chipset</td>
<td>CC2420</td>
</tr>
<tr>
<td>RF frequency (MHz)</td>
<td>2400–2483.5</td>
</tr>
<tr>
<td>Transmit bit rate (kbps)</td>
<td>250</td>
</tr>
<tr>
<td>RX power (mA)</td>
<td>19.7</td>
</tr>
<tr>
<td>TX power (mA)</td>
<td>17.4</td>
</tr>
<tr>
<td>Idle power (µA)</td>
<td>1</td>
</tr>
<tr>
<td>Modulation</td>
<td>DSSS-O-QPSK</td>
</tr>
<tr>
<td>Receive sensitivity</td>
<td>–94 dBm</td>
</tr>
<tr>
<td>Transmission range</td>
<td>Until 100 m</td>
</tr>
<tr>
<td>Multi-channel</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Source: Ceken (2008)

While energy consumption is 15 µJ in sleep mode, it is 35 mJ and 38 mJ in receive mode and transmit mode (Polastre et al., 2004). Receiver and transmission mode energy consumption is 2000 times more than sleep mode energy consumption. So, the wireless sensor sleep mode was ignored. The distance between pixels was considered to be 1 m in the study. The results were reported as percentages.

5.2 Simulation results and discussion

In the experiments, we analysed the scalability of the WSN when varying the total area of the simulation, while maintaining the sensor fixed density. We tested both applications on six different network topologies consisting of 49, 64, 81, 100, 121, and 144 sensor nodes, which are preferred in general (Lino et al., 2011; Gupta et al., 2012). The network of 49 nodes was generated by placing the nodes randomly in a square of 140 × 140 m. Other topologies were generated by scaling the square so that the average node density can remain the same. In the other simulations, packet numbers were they changed to 700, 800, 900, 1000 and similar comparisons were made. The transmission radius of each node was set to 35 m. Experiments were set in static scenario. The initial energy level of the nodes was set to 20 J (Saleem et al., 2012).

The protocol packet delivery ratios are shown in Figure 5. The FP-Ant and BeeWS protocols have the highest packet delivery ratios because both protocols use the flooding method. The AODV and EEABR protocols packet delivery rates dramatically decreased with the growth of the scale. In contrast, the BeeWS maintained a high delivery rate. An important observation is the EEABR packet delivery inconsistency. The reason for this is that the EEABR uses much more control packages. As can be seen, the topology of growth causes instability in the results. In Figure 6, although the FP-Ant and BeeWS protocols have the highest packet delivery ratios, only BeeWS protocol remained stable when the density of packets was increased. Although both protocols use the flooding method, BeeWS protocol reduced double message generation using agents (scouts). Therefore, BeeWS protocol is less affected by the density of packets than other protocols.

Figure 5 Packet delivery ratio comparison according to the number of nodes (see online version for colours)

Figure 6 Packet delivery ratio comparison according to the number of packets (see online version for colours)

The protocol average latency is shown in Figure 7. All protocols have approximately similar average latencies. The BeeWS protocol has the minimum average latency, while the number of nodes is less. All protocols have similar average latency when the number of nodes was increased. Average latency was increased with the number of nodes in topology. In the developed protocol, the dance section that is the route choice section was the most important factor, which increased average latency. However, average latency was less according to other protocols because of the routes stored in a certain period of time. In Figure 8, despite increasing the density of packets, latency increase in the BeeWS and Beesensor protocols were less than other protocols. Because, the BeeWS and Beesensor protocols use agent (scouts). In addition, there was a difference between the BeeWS and Beesensor which results from the number of agents; while the Beesensor uses four different agents (Packers, scouts, foragers and swarms), the BeeWS uses three different agents (Scouts, data transporter and data generator). Therefore, Beesensor generates more control packets than BeeWS.
The protocol energy consumption is shown in Figure 9. While AODV, BeeSensor, and BeeWS protocols have the best energy efficiency as shown in the Figure 9(b), the comparison of all the protocols is shown in Figure 6(a). Although AODV is a protocol that was recognized in literature in terms of energy efficiency, the BeeWS was the most efficient protocol. The reason behind this is that the BeeWS was developed with the honeybee route technique by inspiring the AODV protocol. In addition to this, the BeeWS was more effective than AODV and BeeSensor in terms of energy consumption, since it has a high packet delivery ratio. Energy consumption decreased as long as the loss ratio increased in the system. The FP-Ant and EEBR routing protocols were not shown in the same range with the other three protocols because they had a high-energy consumption.

5.3 Clustering parameters

As previously mentioned, the clustering method was modeled in the DEVS-Sensor environment. In this study, the BeeWS clustering is applied in large-scale WSNs. It was compared to BeeWS non-clustering.

In this set of experiments, we tested both applications on four different network topologies consisting of 144, 256, 400, and 500 sensor nodes. The transmission radius of each member node was set to 35 m. The transmission radius of each cluster head node was set to 105 m. Experiments were set in static scenario. The initial energy levels of the nodes were set at 20 J. The event in application is 512-bits long and generated at a rate of one event per second. Each experiment was performed for a duration of 300 s, and the reported values were an average of 10 independent runs. Cluster head reached the destination node using the method of multi-hop. Evaluation purposes, packet delivery ratio, average latency, and energy consumption criteria were used. Also, experiments were conducted on the number of clusters and cluster heads. Experiments were performed till 2000 nodes. It is compared EECF and EESH clustering protocols.

Figure 7 Average latency comparison according to the number of nodes (see online version for colours)

Figure 8 Average latency comparison according to the number of packets (see online version for colours)

Figure 9 Comparison of energy consumption (see online version for colours)

The packet delivery rate is shown in Figure 10. Packet delivery rate tended to reduce despite the fact that the number of sensor node increased in BeeWS non-clustering. The reason for this is that the ability of finding a route decreased due to the increase in the number of nodes in the topology. The number of member nodes decreased in topology using cluster head via the BeeWS clustering method. Almost all of the packets were delivered to their destination. The BeeWS non-clustering method was not applied to more than 500 nodes. The packet delivery rate was high due to instability after 400 nodes in the simulation tool that applied the BeeWS non-clustering.
Average latency is shown in Figure 11. The number of nodes increased in the topology as well as the number of events increased. Accordingly, average latency increased depending on high traffic in the BeeWS non-clustering. The WSN traffic reduced and the average latency decreased because the number of member nodes was decreased in the topology using the cluster head via the BeeWS clustering method.

Energy consumption is shown in Figure 12. Gain in energy consumption was obtained using the method of clustering. Certain nodes’ energy was consumed because all nodes do not take part in clustering. The WSN lifetime was extended because the cluster heads that decreased energy changed with member nodes. This situation also enabled the use of energy in a controlled manner. It demonstrated a big difference in terms of energy consumption between the BeeWS clustering and BeeWS non-clustering in Figure 12.

The number of average cluster heads is shown in Figure 13. The number of cluster heads are remained stable about 5% when the number of nodes increases. LEACH clustering method that is the most cited in the literature determined as 5% the number of cluster heads. Therefore, BeeWS clustering method is similar to other clustering methods accepted in the literature. It is compared in terms of number of clusters with EEHC and EESH in Figure 14. The number of clusters affect energy consumption. When the number of clusters are increased, the number of selected cluster heads are also increased. This results in more energy consumption and disrupt the energy balance in WSN. BeeWS clustering constitutes less clusters than the other methods and the number of clusters were remained stable about 5%.

6 Conclusion

In this paper, we have proposed a new routing protocol, the BeeWS, which is large-scale and based on SI inspired by honeybees for WSNs. Sensor nodes, like honeybees, can scale by being self-organised and create a intelligent system, as BeeWS is based on SI. Experimental results have shown that the BeeWS has better performance than other well-known routing protocols in terms of packet delivery ratio, energy consumption, and latency. The reason for this is that

- we inspired AODV developed for ad-hoc network
- we used control packets as often as necessary
the routes that were founded by the BeeWS protocol were temporarily stored in the sensor nodes
the most convenient route was selected among the routes found
network lifetime was prolonged by monitoring all network energy during the process of route finding
when the number of sensor nodes increased, the WSN was divided into autonomous systems by clustering.

Therefore, energy consumption was efficient in the WSN and decreased the packet load. In future work, we will aim to apply the BeeWS protocol to bigger WSNs and mobile scenarios.

References
