Biologically-inspired Adaptive Power Management for Wireless Sensor Networks

Pruet Boonma and Junichi Suzuki
Department of Computer Science
University of Massachusetts, Boston
{pruet, jxs}@cs.umb.edu

Abstract

This paper describes BiSNET (Biologically-inspired architecture for Sensor NETworks), which addresses a key issue in wireless sensor networks (WSNs): autonomous, scalable, adaptive and simple power management. Based on the observation that various biological systems have developed mechanisms to achieve autonomy, scalability and adaptability based on a set of simple principles, BiSNET implements certain biological principles and mechanisms to design sensor network applications. This paper presents the biologically-inspired principles and mechanisms in BiSNET, and evaluates their impacts on the power management of WSNs. Simulation results show that BiSNET allows each sensor node to autonomously operate and adaptively perform power management according to dynamic changes in node status and network conditions. Simulation results also demonstrate that BiSNET scale well to network size and transmitted data volume. The BiSNET runtime is designed simple and implemented lightweight.

1. Introduction

One of the critical issues in battery-operated wireless sensor networks (WSNs)[1] is power efficiency. This paper addresses four major requirements to the power efficiency issue. The first requirement is *autonomy*. Since sensor nodes can be deployed in an unattended area (e.g., forest and ocean), they are required to manage their power consumption with the minimum aid from base stations or human administrators [1, 2].

The second requirement is *scalability*. In order to cover large spatial extents or monitor the extents at a high temporal resolution, sensor networks are required to retain their power efficiency against a large number of sensor nodes[2] and a large amount of data generated by sensor nodes [1, 3].

The third requirement is *adaptability*. Sensor nodes are required to adapt their power management operations to the environmental conditions that they monitor (e.g., temperature and carbon monoxide (CO)) [3, 4, 5]. For example, sensor nodes may increase their sleep periods when there is no significant change in their sensor readings. This results in less

[1] This paper considers multi-modality in WSNs. Each sensor network deploys multiple types of sensor nodes in an observation area (e.g., temperature, humidity and carbon monoxide sensor nodes). Data from different types of sensor nodes are aggregated, through in-network processing, to provide a multi-dimensional view of observed environmental conditions.

[2] For example, the DARPA Networked Embedded Systems Technology program envisions networks consisting of 100 to 100,000 sensor nodes.

power consumption in the nodes. Sensor nodes may aggregate data from different types of sensor nodes (e.g., temperature and CO data) and transmit the aggregated data to a base station. Sensor nodes may also vary data transmission paths to a base station when they transmit sensor data to the base station very often. The data aggregation and transmission path adjustment can reduce power consumption in the nodes on the paths toward a base station. This avoids a network to be separated into islands by the data transmission paths on which too many data travel.

The fourth requirement is *simplicity*. Due to limited resource availability, sensor control software needs to be simple in its design and lightweight in its footprint in order to minimize power consumption.

This paper describes an architecture for power efficient WSNs, called BiSNET (Biologically-inspired architecture for Sensor NETworks), which addresses the above four requirements. BiSNET is motivated by the observation that various biological systems have already developed mechanisms to meet those requirements [6, 7]. For example, bees act autonomously, influenced by local conditions and local interactions with other bees. A bee colony can scale to a massive number of bees because all activities of the colony are carried out without centralized control. A bee colony adapts to dynamic environmental conditions. For example, when the amount of honey in a hive is low, many bees leave the hive to gather nectar from flowers. When the hive is full of honey, bees rest in the hive or expand the hive. The structure and behavior of each bee are very simple; however, a group of bees autonomously exhibits desirable system characteristics such as scalability and adaptability through collective behaviors and interactions among bees. Based on this observation, the authors of the paper believe that, if WSN applications are designed after certain biological principles and mechanisms, they may be able to attain the requirements in WSNs (i.e., autonomy, scalability, adaptability and simplicity).

The BiSNET runtime operates atop of TinyOS in each sensor node (Figure 1). It consists of a middleware platform and one or more agents. BiSNET models a platform as a hive and agents as bees. Agents are designed to follow several biological principles such as decentralization, autonomy, food gathering/consumption and natural selection. Each agent reads sensor data with the underlying sensor devise, and discards or reports it to a base station using biological behaviors such as energy exchange, pheromone emission, replication and migration. Each platform runs on TinyOS and hosts agents. It controls the state of a sensor node (e.g., sleep, listen and broadcast), and provides a set of runtime services that agents use to read sensor data and perform their behaviors.

This paper describes the biologically-inspired mechanisms in BiSNET and evaluates their impacts on the autonomy, scalability, adaptability and simplicity of power management in WSNs. Simulation results show that BiSNET allows sensor nodes (agents) to autonomously improve power efficiency by adaptively adjusting their sleep periods (duty cycles), aggregating data from different types of nodes and varying data transmission paths. Simulation results also show that BiSNET allows sensor nodes (agents) to scale and retain their power efficiency against the increase of network size and data volume. The BiSNET runtime is implemented lightweight thanks to a set of simple biological mechanisms.

2. Design of BiSNET

This section describes the design of agents and platforms in BiSNET.

2.1. Design Principles for BiSNET Agents

(1) Decentralization: Inspired by biological systems (e.g., bee colonies), there are no centralized entities in BiSNET to control and coordinate agents. Decentralization allows agents to be scalable and simple by avoiding a single point of performance bottlenecks and failures [8, 9] and by avoiding any central coordination in deploying agents [10].

(2) Autonomy: Similar to biological entities (e.g., bees), agents sense their local environments, and based on the sensed conditions, autonomously behave without intervention from/to other agents, platforms, base stations and human operators.

(3) Food gathering and consumption: Biological entities strive to seek and consume food for living. For example, bees gather nectar from flowers and digest it to produce honey. In BiSNET, agents (bees) read sensor data (nectar), and digest it to *energy* (honey)[3]. (Energy gain is proportional to a change between the current and previous sensor data.)

(4) Natural selection: The abundance or scarcity of stored energy in agents affects their behaviors and triggers natural selection. For example, an energy abundance indicates a significant change in sensor reading; thus, an agent emits a *pheromone* to stimulate replicating itself and its neighboring agents. A replicated agent migrates to a neighboring node to report sensor data to a base station. An energy scarcity (an indication of few changes in sensor reading) eventually causes the death of agents. Like in biological natural selection where more favorable species in the environment becomes more abundant, the population of agents dynamically changes based on their energy levels (i.e., changes in their sensor readings).

2.2. BiSNET Agent

Each agent consists of *attributes*, *body* and *behaviors*. *Attributes* carry descriptive information on an agent. They include agent type (e.g., temperature sensing agent and CO sensing agent), energy level, sensor data to be reported to a base station, time stamp of the sensor data, and ID/location of a sensor node where the sensor data is captured. Application developers can define arbitrary attributes for their agents.

Body implements the functionalities of the agent: collecting and processing sensor data. In each duty cycle, each agent gathers sensor data (as food) from the underlying sensor device, converts it to energy and processes it (e.g., discards it or reports it to a base station). Different types of agents collect different types of sensor data.

Behaviors implement actions inherent to all agents. This paper focuses on the following five behaviors.

[3]The concept of energy in BiSNET does not represent the amount of physical battery in a node. It is a logical concept that affects agent behaviors.

- *Pheromone emission:* Agents may emit different types of pheromones (*replication pheromones* and *migration pheromones*) according to their local and surrounding network conditions. Agents emit replication pheromones in response to the abundance of stored energy (i.e., significant changes in their sensor readings). Different types of agents emit different types of replication pheromones, each of which carries sensor data. For example, temperature sensing agents emit temperature pheromones, which carry temperature data. CO sensing agents emit CO pheromones, which carry CO data. Replication pheromones stimulate the agents on the local and neighboring nodes to replicate themselves. Each replication pheromone can spread to one-hop away neighboring sensor nodes. On the other hand, agents emit migration pheromones on their local nodes when they migrate to neighboring nodes. Each replication and migration pheromone has its own concentration (or strength). The concentration decays by half at each duty cycle. A pheromone disappears when its concentration becomes zero.
- *Replication:* Agents may make a copy of themselves in response to the abundance of energy and replication pheromones. Each agent does not initiate replication until enough types of replication pheromones become available on the local node. For example, an agent may replicate itself only when both temperature pheromones and CO pheromones are available. A replicated (child) agent retains the same agent type as its parent's type, and aggregates multiple sensor data stored in multiple types of available replication pheromones. A child agent is placed on the node that its parent agent resides on, and it receives the half amount of the parent's energy level. Each child agent is intended to move toward a base station to report (aggregated) sensor data.
- *Migration:* Agents may move from one sensor node to another in response to energy abundance (i.e., significant changes in their sensor readings). Migration is used to transmit agents (sensor data) to base stations. Each agent may implement one of or a combination of the following four migration policies:
  - **Directional walk:** Each agent may move to the nearest base station through the shortest path. Each base station periodically propagates *base station pheromones*, whose concentration decays on a hop-by-hop basis. Using base station pheromones, agents can sense where base stations exist approximately, and move toward the base stations by climbing pheromone gradients.
  - **Chemotaxis:** Agents may move to base stations by following migration pheromone traces on which many other agents travel. These traces can be the shortest paths to the base stations. When there are no migration pheromones on neighboring nodes, agents perform directional walk.
  - **Detour walk:** Each agent may go off a migration pheromone trace and follows another path to a base station when the concentration of migration pheromones is too high on the trace (i.e., when too many agents follow the

same migration path). This avoids separating the network into islands. The network can be separated with the migration paths that too many agents follow, because the nodes on the paths consume more power than others and they go down earlier than others. In addition to the detour with migration pheromones, agents may avoid moving through the nodes where the concentration of replication pheromones is too high (i.e., where agents detect significant changes in their sensor readings). This detour walk distributes power consumption of agent migration over the nodes where agents do not detect no changes in their sensor readings, thereby avoiding the network to be separated.

- *Energy exchange:* Agents periodically deposit some of their energy units (honey) to their local platforms (hives), and keep the rest for living (i.e., for invoking their behaviors).
- *Death:* Agents die due to lack of energy when they cannot balance energy gain and expenditure. When an agent dies, the local platform removes the agent and releases all resources allocated to the agent.

Every agent expends a certain amount of energy to perform replication and migration behaviors. The energy costs to invoke these behaviors are constant for all agents.

Figure 2 shows a sequence of actions that each agent performs in each duty cycle. First, an agent reads sensor data (as nectar) with the underlying sensor device, and converts it to energy (honey). The energy intake (E_F) is calculated with Equation 1. S represents the absolute difference between sensor data in the current and previous duty cycle. M is the metabolic rate, which is a constant value between 0 and 1.

$$E_F = S \cdot M \tag{1}$$

Different platforms may have different M values to prioritize particular types of sensor nodes. All agents on a platform follow the same M value that the platform has. The higher M value a platform has, the more often agents replicate and migrate on the platform because of higher energy intake.

Given E_F, each agent updates its energy level as follows.

$$E(t) = E(t-1) + E_F \tag{2}$$

$E(t)$ is the current energy level of the agent, and $E(t-1)$ is the agent's energy level in the previous duty cycle. t is incremented by one at each duty cycle. Note that agents always exchange and share their energy units equally with other agents in the same node.

If an agent's energy level ($E(t)$) becomes very low (below the death threshold: T_D), the agent dies due to energy starvation (see also Figures 2 and 3)[4].

[4] If all agents are dying on a node at the same time, a randomly selected agent will survive. At least one agent runs on each node.

Then, an agent emits a replication pheromone if its energy level exceeds its replication pheromone emission threshold T_P (see Figures 2 and 3). Agents continuously adjust their replication pheromone emission thresholds as the EWMA (Exponentially Weighted Moving Average) of their energy levels:

$$T_P(t) = (1-\alpha)T_P(t-1) + \alpha E(t) \tag{3}$$

$T_P(t)$ is the current replication pheromone emission threshold, and $T_P(t-1)$ is the one in the previous duty cycle. EWMA is used to smooth out short-term minor oscillations in the data series of E (energy level of an agent). It places more emphasis on the long-term transition trend of E; only significant changes in E have the effects to change T_P. The α value is a constant to control the sensitivity of T_P against the changes of E.

When a replication pheromone is emitted on a node, all the agents on the node can sense it. It may stimulate their replications. An agent replicates itself when it meets two condition: (1) when the agent's energy level ($E(t)$) exceeds its replication threshold (T_R), and (2) when the concentration of each type of available replication pheromones (P_i[5]) exceeds its stimulation threshold T_{S_i} (see Figures 2 and 3). Agents continuously adjust their replication thresholds as the EWMA of their energy levels (Equation 4). The stimulation threshold of a replication pheromone changes as the EWMA of the pheromone's concentration (Equation 5).

$$T_R(t) = (1-\beta)T_R(t-1) + \beta E(t) \tag{4}$$

$$T_{S_i}(t) = (1-\gamma)T_{S_i}(t-1) + \gamma P_i(t) \tag{5}$$

$T_R(t)$ is the current replication threshold, and $T_R(t-1)$ is the one in the previous duty cycle. T_{S_i} is the current replication pheromone stimulation threshold for the replication pheromone type i, and $T_{S_i}(t-1)$ is the one in the previous duty cycle. The β and γ values are the constants to control the sensitivity of T_R and T_{S_i} against the changes of E and P_i, respectively. A replicating (parent) agent splits its energy units to halves ($\frac{E(t)-E_R}{2}$), gives a half to its child agent, and keeps the other half. E_R is the cost (energy units) for an agent to invoke the replication behavior. A replicated (child) agent aggregates the sensor data in the pheromones that stimulated its parent agent to perform a replication. A parent agent keeps replicating itself until its energy level becomes less than its replication threshold (T_R). Replicated agents may migrate to neighboring nodes when the local node is in broadcasting state (see Figure 2).

As described above, agents replicate themselves only when they gain a large amount of energy on the local node and receive enough types of high-concentration pheromones from the local and neighboring nodes. This means that sensor data are aggregated and transmitted to base stations only when significant changes in sensor data are detected on the local and

[5] P_i denotes the total concentration of replication pheromone type i. i is used to indicate different types of replication pheromones available on the local node (e.g., temperature and CO pheromones).

neighboring nodes. Agents do not respond to gradual changes in sensor readings (e.g., temperature change during a day or between different seasons). This reduces power consumption in sensor nodes and expands the life of a WSN by avoiding unnecessary data transmission.

This adaptive data aggregation and transmission mechanism are designed with a self-healing capability in mind, which allows agents to detect and eliminate false positive sensor data. When a sensor node does not work properly due to, for example, malfunctions or miscalibrations, each agent on the node emits the replication pheromones that contain false positive sensor data. A large number of false positive pheromones may be transmitted to neighboring nodes. However, they are discarded at the neighboring nodes because they are not aggregated with other types of pheromones (see Figure 2). This means that false positive pheromones are not propagated more than two hops from a malfunctioning or miscalibrated node. Also, agents stop emitting false positive pheromones on the malfunctioning/miscalibrated node because their pheromone emission thresholds increase (see Equation 3).

Each agent deposits a certain amount of energy (E_P) to the platform that it resides on (see also Figures 2 and 3):

$$E_P = \begin{cases} E(t) - E(t-1) & \text{if } E(t) \geq E(t-1) \\ 0 & \text{if } E(t) < E(t-1) \end{cases} \tag{6}$$

Each agent strives to keep its energy level ($E(t)$) close to the one in the previous duty cycle ($E(t-1)$).

When a platform's total energy gain ($\sum E_P$) is greater than a threshold (T_B), the platform changes its state to the broadcast state. This allows replicated agents and pheromones to move to neighboring nodes (see Figures 2 and 3). Each agent implements one of or a combination of three migration policies (directional walk, chemotaxis and detour walk; see Section 2.2) with the following equation.

$$WS_j = \sum_{t=1}^{3} w_t \frac{P_{t,j} - P_{t_{min}}}{P_{t_{max}} - P_{t_{min}}} \tag{7}$$

Each agent calculates this weighted sum (WS) for each neighboring node j, and moves to a node that generates the highest weighted sum. t indicates pheromone type; P_{1j}, P_{2j} and P_{3j} represent the concentration of base station, migration or replication pheromones on a neighboring node j. $P_{t_{max}}$ and $P_{t_{min}}$ are the maximum and minimum concentration of P_t among neighboring nodes. w_t is used to determine which migration policies each agent performs. w_2 and w_3 are zero for agents performing directional walk. w_2 is positive and negative for agents performing chemotaxis and the detour walk with migration pheromones, respectively. w_3 is negative for agents performing the detour walk with replication pheromones.

2.3. BiSNET Platform

Each platform consists of two parts: *runtime services* and *state controller*. *Runtime services* hide lower-level computing and networking details (e.g., network I/O), and provide high-level services that agents use to read sensor data and perform behaviors (see also Figure 1). For example, the runtime services allow each agent to sense the type and concentration of each pheromone available on the local node.

State controller dynamically changes the state of a sensor node to control its duty cycle (sleep period). Each sensor node is in the *listen*, *broadcast* or *sleep* state (Figure 4). A platform and agents can work on a sensor node when its state is in the listen or broadcast state. In the listen state, a platform turns on a radio receiver to receive data (agents and pheromones) from neighboring sensor nodes. Each agent performs a series of actions described in Figure 2. The listen state changes to the broadcast state if a platform gains energy more than the broadcast threshold ($\sum E_P > T_B$; see also Figures 3 and 4). In the broadcast state, a platform turns on a radio transmitter to allow agents and pheromones to move to neighboring nodes.

When a platform gains no energy from agents ($\sum E_P = 0$), the platform goes into the sleep state (Figure 4). The sleep period is determined as follows. P_{sleep} is a constant, and P_i is the concentration of each type of replication pheromones (the pheromone type i) available on the platform.

$$\text{sleep period} = \begin{cases} \frac{P_{\text{sleep}}}{\sum P_i} & \text{if } \sum P_i > 0 \\ P_{\text{sleep}} & \text{if } \sum P_i = 0 \end{cases} \tag{8}$$

The sleep period is reverse proportional to the total concentration of replication pheromones available on a platform ($\sum P_i$). This means that a platform increases its sleep period to reduce power consumption when agents find no significant changes in their sensor readings on the platform and its neighboring platforms.

This adaptive duty cycle management mechanism is designed with an inference capability in mind. When a platform receives replication pheromones from a neighboring node(s), it decreases its sleep period even if there is no change in the sensor reading on the local node (see Equation 8). This way, agents can be more watchful on the node for a future potential change in their sensor readings so that they do not miss it during sleep period.

3. Simulation Results

This section shows a series of simulation results to evaluate how the biologically-inspired mechanisms in BiSNET impact the autonomy, adaptability, scalability and simplicity of power management in WSNs.

3.1. Simulation Configurations

This simulation study emulates a WSN deployed in a forest to detect wildfires. The WSN consists of temperature sensor nodes and CO sensor nodes randomly deployed in a 25x24 grid topology (600 nodes); a half of the nodes equip temperature sensors, and the other half equip CO sensors (Figure 5). A wildfire moves from southeast to northwest. Simulations follow a model that describes wildfire spreading in nature [11]. The following five BiSNET configurations are evaluated to examine how different biologically-inspired mechanisms in BiSNET impact the operation of WSNs.

- **BiSNET-Mb:** Agents do not perform the replication behavior with replication pheromones. They replicate themselves when their energy levels exceed their replication thresholds (T_R), and do not perform data aggregation. Agents migrate to the base station with base station pheromones (directional walk; see Section 2.2).
- **BiSNET-RMb:** Agents perform the replication behavior with replication pheromones; they perform data aggregation. Agents migrate to the base station with base station pheromones (directional walk; see Section 2.2).
- **BiSNET-RMbm:** Agents perform the replication behavior (data aggregation) with replication pheromones. They migrate to the base station with base station and migration pheromones (directional walk, chemotaxis and detour walk with migration pheromones; see Section 2.2). They perform chemotaxis by default and execute detour walk when the concentration of migration pheromones is too high on their local nodes.
- **BiSNET-RMbr:** Agents perform the replication behavior (data aggregation) with replication pheromones. They migrate to the base station with base station and replication pheromones (directional walk and detour walk with replication pheromones; see Section 2.2).
- **BiSNET-RMbmr:** Agents perform the replication behavior (data aggregation) with replication pheromones. They migrate to the base station with all of three pheromones (directional walk, chemotaxis, and detour walk with migration and replication pheromones; see Section 2.2).

In addition, BiSNET is compared with following four existing routing protocols for WSNs.

- **GBR (Gradient Based Routing):** In GBR [12], a base station periodically propagates a routing message to sensor nodes throughout the network. The routing message gradually assigns smaller *gradient hight* values to nodes as it travels on a hop by hop basis. Given a gradient toward a base station, each node forwards sensor data to a neighboring node that has a higher gradient hight. This routing protocol is similar to BiSNET-Mb. In GBR, sensor data are transmitted on the shortest paths to base stations; it is likely that network separations occur. To alleviate this problem, there are three variations of GBR [12]:
  - **GBR-R:** When a node finds multiple neighboring nodes that have the same gradient hight (i.e., the same distance to a base station), it randomly selects one of them and forwards sensor data to the selected node.
  - **GBR-P:** When the remaining amount of power becomes low on a node, the node increases its gradient hight so that it does not receive sensor data from neighboring nodes.
  - **GBR-S:** Nodes divert data transmission paths (or streams) to base stations. When a node receives a sensor data from a neighboring node, it increases its gradient hight for a while so that it does not receive sensor data from neighboring nodes. This routing protocol is similar to BiSNET-Mbm.

3.2. Success Rate and Latency of Data Transmission

Table 1 shows the total number of sensor data that nodes collect and report to the base station throughout a simulation. BiSNET-RMb, -RMbm, -RMbr and -RMbmr collect/report more sensor data than BiSNET-Mb because of data aggregation based on replication pheromones. The increase in the number of collected data is approximately 16%. Compared with GBR, BiSNET always operate in a higher temporal resolution because the inference mechanism in BiSNET allows nodes to collect more sensor data by reducing sleep periods. BiSNET-RMbm, -RMbr and -RMbmr collect 22% more data than any GBR protocols do. Please note that the success rate of data transmission is almost same between BiSNET and GBR even if BiSNET collects and report more data to the base station.

Table 2 shows the average latency to transmit sensor data from nodes to the base station. With data aggregation enabled with replication pheromones, the latency is shorter in BiSNET-RMb than BiSNET-Mb because data aggregation reduces the number of migrating agents, and in turn, network traffic. In BiSNET-RMbm, -RMbr and -RMbmr, agents perform detour walk and do not always travel on the shortest path to the base station; however, the latency is not severely affected. The latency of BiSNET-RMbmr is only one second longer (or 3% longer) than that of BiSNET-Mb. Compared with GBR, the latency of BiSNET is one to three seconds longer (or 3% to 9% longer) because BiSNET transmits more data than GBR as shown in Table 1. The authors of the paper believe that the increase of 3% to 9% in latency is acceptable against the increase of 22% in data collection. Note that the standard deviation of latency is almost same in BiSNET and GBR.

3.3. Power Consumption

Table 3 shows the average power consumption of sensor nodes throughout a simulation. In BiSNET-RMb, -RMbm, -RMbr and -RMbmr, with data aggregation enabled with replication pheromones, agents reduce the number of transmitted data to the base station via data aggregation. Compared with BiSNET-Mb, this contributes to 3% to 6% reduction in average power consumption. The power consumption in BiSNET-RMbm, -RMbr and -RMbmr is 2% to 3% higher than BiSNET-RMb because agents perform detour walk and do not always travel on the shortest path to the base station. Compared with GBR, BiSNET consumes 10% to 16% more power beause it transmits more data as shown in as shown in Table 1. However, the power consumption per reported sensor data is consistently lower in BiSNET than GBR. For example, BiSNET-RMbmr consumes 12% to 18% less power per reported data than GBR-R, -P and -S. This means that BiSNET manages power consumption effectively while increasing the temporal resolution of data collection. Please also note that the standard deviation of power consumption is consistently lower in BiSNET than GBR. This means BiSNET distributes power consumption over more nodes, thereby reducing a risk of network separation more effectively than GBR.

Figures 6, 7, 8, 9 and 10 show how much power is consumed on 600 (25x24) nodes in five BiSNET configurations.

In each figure, the base station is located at the upper corner of the network. A wildfire starts at the bottom corner of the network, and move upward. These figures show that BiSNET reduces and distribute power consumption over nodes by leveraging data aggregation (with replication pheromones) and detour walk (with migration and replication pheromones). These mechanisms contribute to reduce a risk of network separation and expand the network lifetime.

3.4. Network Lifetime

Figure 11 shows how soon sensor nodes go down due to lack of power in different BiSNET and GBR configurations. In BiSNET-Mb, 64 of 600 nodes (11% nodes) go down in 1,000 minutes. With data aggregation enabled with replicaiton pheromones, 57 nodes (9.5% nodes) go down in 1,000 minutes. Moreover, with detour walk with migration and replication pheromones, the number of dead nodes decreases to 47 nodes (7.8% nodes) in BiSNET-RBbmr. Only one node goes down in 500 minutes. In contrast, in GBR-S, three nodes go down in 500 minutes and 52 nodes (8.7% nodes) in 1,000 minutes. The first death of sensor node occurs in 482 minutes in BiSNET-RBbmr and in 350.5 minutes in GBR-S. BiSNET-RBbmr delays the first node death by 131.5 minutes (more than two hours). BiSNET successfully reduces and distributes power consumption over nodes and expand the network life[6].

3.5. Scalability

In order to evaluate the scalability of BiSNET against the number of nodes, the same set of simulations was carried out on the WSNs consisting of 600 nodes and 56 nodes. The simulation results in the case of 56 nodes are not presented due to space limitation; however, the authors of the paper have confirmed that the results are qualitatively same in the cases of 600 nodes and 56 nodes. Simulation results show that BiSNET is scalable against the increase of network size and data volume generated by nodes.

3.6. Simplicity: Memory Footprint

In order to evaluate the simplicity of BiSNET, Table 4 shows the memory footprint of the BiSNET runtime in a MICA2 mote, and compares it with the footprint of Blink (an example program in TinyOS), which periodically turns on and off an LED, GBR, and Agilla, which is a mobile agent platform for WSNs [15]. As shown in Table 4, the BiSNET runtime is fairly lightweight in its footprint, and it can be deployed on sensor devises whose resource availability is severely limited.

4. Related Work

In the previous work of the authors of the paper [16, 17], agents in BiSNET migrate to a base station on a shortest-path basis only with base station pheromones. In this work, BiSNET is extended to support multiple types of pheromones, i.e.

[6]GBR outperforms the directed diffusion protocol [13] in terms of power efficiency [14]. Thus, it is fair to say that BiSNET-RBbmr also outperforms the directed diffusion protocol.

base station pheromones, replication pheromones and migration pheromones, and improve power efficiency of WSNs via adaptive migration path adjustment. As simulation results show, the network lifetime is expanded via power efficiency while retaining the performance in latency and success rate of data transmissions.

There are several research efforts to apply biological mechanisms to sensor networks. For example, in order to synchronize clocks of sensor nodes in a decentralized manner, [18] applies firefly's phase synchronization mechanism in which fireflies synchronize their light on/off periods with each other. BiSNET focuses on different issues; it applies biological mechanisms to adaptive duty cycle management, inference on potential environmental changes and data aggregation and self-healing of false positive data.

[19] proposes to apply biological mechanisms to an operating system for sensor networks, called kOS, in order to make them robust to topological changes, scalable and self-organizing. However, kOS has not implemented any specific biological mechanisms yet. In contrast, BiSNET specifically implements biological mechanisms such as energy exchange, pheromone emission, replication, migration and death to improve the ability of sensor nodes for power efficiency, inference and self-healing.

Agilla proposes a programming language to implement mobile agents for sensor networks, and provides a runtime system (interpreter) to operate agents on TinyOS [15]. BiSNET does not focus on investigating a new programming language for sensor networks. BiSNET agents and Agilla agents have a similar set of behaviors such as migration and replication. Both of them are also intended to be used for similar applications (e.g., wildfire detection). However, Agilla does not address the research issues that BiSNET focuses on; power efficiency, data aggregation, inference and self-healing. In addition, BiSNET focuses on its design simplicity and runtime lightweightness. As shown in Figure 4, BiSNET is much more lightweight than Agilla.

[20, 21, 22] describe dynamic duty cycle management in sensor nodes. Their goal is to improve power efficiency, and they do not consider sensing responsiveness for potential environmental changes (i.e., the risk to miss significant environmental changes during sleep period). Unlike them, the duty cycle management scheme in BiSNET is designed to adaptively balance the tradeoff between power efficiency and sensing responsiveness for potential environmental changes. As a result, BiSNET uses sensor data (i.e., pheromones) to determine the sleep period of each sensor node, while [20, 21, 22] randomly change sleep period or use other metrics such as the average time for a sensor node to process packets.

Quasar proposes a data collection protocol that balances the tradeoff between data accuracy and power efficiency [23]. In Quasar, each sensor node switches its state between active and idle (sleep) to minimize its power consumption. A central server controls the periods of active and idle states based on the changes in sensor readings. Unlike Quasar, BiSNET does not require any central server; individual sensor nodes locally adjust their duty cycle intervals. In addition, BiSNET implements two ways to trigger dynamic duty cycle adjustment: based on changes in sensor reading on the local node and via inference from sensing activities of neighboring nodes. Quasar does not implement the inference function.

[24] proposes a middleware platform for MWSNs. It is implemented with a scripting language (Python) to improve the ease of developing applications. In contrast, BiSNET is implemented with NesC, instead of a scripting language, not to sacrifice its performance and runtime lightweightness. [24] allows each sensor node to aggregate different types of sensor data. Application programmers are required to explicitly specify (or hard code) the condition to aggregate sensor data at each node (e.g., temperature > 200 and CO level > 200). In BiSNET, application programmers do not have to specify data aggregation condition for each node. Rather than using hard-coded data aggregation conditions, each node aggregates sensor data generated by the node and its neighboring nodes when the sensor data change significantly (see Figure 2 and Equations 4 and 5). In addition, [24] does not consider the issues that BiSNET focuses on, such as adaptive duty cycle management for power efficiency, inference on potential environmental changes for sensing responsiveness and self-healing of false positive sensor data.

[25] proposes to divide the functionality of a wireless sensor network into three parts: communication, collaborative sensing and operational commands. Each part has a Petri Net to control the behavior of each sensor node. [25] is similar to BiSNET in that it can detect and eliminate false positive data as the function of collaborative sensing. However, BiSNET is designed much simpler; it has only a few states and transitions among them, while [25] has 133 states and 232 transitions among them. [25] does not consider design simplicity and runtime lightweightness. In addition, BiSNET operates in a decentralized manner, while [25] organizes a sensor network in a hierarchical manner.

In addition, the pheromone based communication primitive in this work does not provide any direct benefit; application developers have to carefully design their own application to gain benefits from pheromone mechanism. In BiSNET, the data aggregation, inference, and adaptive sleep period are direct benefits from using pheromone, application developers do not need to concern those issues when using BiSNET.

5. Concluding Remarks

This paper describes a biologically-inspired architecture, called BiSNET, which coherently applies a small set of simple biological mechanisms to improve the autonomy, scalability, adaptability and simplicity of power efficient WSN applications. BiSNET simultaneously provides simple yet generic solutions for the design issues in power efficient WSNs (duty cycle management, data aggregation, data transmission path adjustment, self-healing and inference) rather than focusing on them one by one or in an ad-hoc manner. Simulation results show that BiSNET allows sensor nodes (agents) to autonomously improve power efficiency by adaptively adjusting their sleep periods, aggregating data from different types of nodes and varying data transmission paths. Simulation results also show that BiSNET allows sensor nodes (agents) to scale and retain their power efficiency against the increase of network size and data volume. The BiSNET runtime is implemented lightweight thanks to a set of simple biological mechanisms.

Several extensions to BiSNET are planned. One of the key extensions is to deploy BiSNET on real WSNs, beyond simulation studies, in collaboration with marine scientists and ecologists of the Center for Coastal Environmental Sensing Networks at the University of Massachusetts Boston. BiSNET is planned to be used to monitor episodic events (e.g., spills and harmful algae blooms) in shallow-water coastal environments.

References

[1] M. Yu, H. Mokhtar, and M. Merabti, "A survey of network management architecture in wireless sensor network," in *Proc. of the Sixth Annual PostGraduate Symposium on The Convergence of Telecommunications, Netowrking & Broadcasting*, Jun 2006.

[2] J. Blumenthal, M. Handy, F. Golatowski, M. Haase, and D. Timmermann, "Wireless sensor networks - new challenges in software engineering," in *Proc. of Emerging Technologies and Factory Automation*, September 2003.

[3] S. Hadim and N. Mohamed, "Middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 853–865. IEEE, 2006.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Elsevier Journal of Computer Networks*, vol. 38, pp. 393–422, 2002.

[5] J. Ledlie, J. Schneidman, M. Welsh, M. Roussopoulos, and M. Seltzer, "Open problems in data collection networks," in *Proc. of European SIGOPS*, September 2004.

[6] D. Gordon, *Ants At Work: How an Insect Society is Organized*. The Free Press, 1999.

[7] T. Seeley, *The Wisdom of the Hive*. Harvard University Press, 2005.

[8] R. Albert, H. Jeong, and A. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, July 2000.

[9] N. Minar, K. H. Kramer, and P. Maes, "Cooperating mobile agents for dynamic network routing," *Software Agents for Future Communications Systems*. Springer, 1999.

[10] G. Cabri, L. Leonardi, and F. Zambonelli, "Mobile-agent coordination models for internet applications," *IEEE Computer*, February 2000.

[11] B. Drossel and F. Schwabl, "Self-organized critical forest-fire model," *Phys. Rev. Lett.*, vol. 69, no. 11, pp. 1629–1632. American Physical Society, Sep 1992.

[12] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Proc. of MILCOM 2001*, 2001.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. of the 6th ACM/IEEE Int'l COnference on Mobile Computing and Networking*, August 2000.

[14] J. Faruque, K. Psounis, and A. Helmy, "Analysis of gradient-based routing protocols in sensor networks," in *Proc. of IEEE/ACM Int'l Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2005.

[15] C.-L. Fok, G.-C. Roman, and C. Lu, "Rapid development and flexible deployment of adaptive wireless sensor network applications," in *Proc. of Int'l Conf. on Distributed Computing Systems*, June 2005.

[16] P. Boonma, P. Champrasert, and J. Suzuki, "Bisnet: A biologically-inspired architecture for wireless sensor networks," in *Proc. of The Second IEEE International Conference on Autonomic and Autonomous Systems*, July 2006.

[17] P. Boonma and J. Suzuki, "Biologically-inspired data aggregation for multi-modal wireless sensor networks," in *Proc. of The 31th IEEE Conference on Local Computer Network (LCN)*, November 2006.

[18] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. of The Third Int'l Conf. on Embedded Networked Sensor Systems*, 2005.

[19] M. Britton, V. Shum, L. Sacks, and H. Haddadi, "A biologically inspired approach to designing wireless sensor networks," in *Proc. of The Second European Workshop on Wireless Sensor Networks*, 2005.

[20] C.-F. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms," in *Proc. of The Third Int'l Symposium on Information Processing in Sensor Networks*, 2004.

[21] J. Mišić and V. B. Mišić, "Duty cycle management in sensor networks based on 802.15.4 beacon enabled MAC," *Ad Hoc and Sensor Wireless Networks Journal*, vol. 1, no. 3, pp. 207–233. Old City Publishing, Inc., 2005.

[22] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM trans. on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[23] Q. Han, S. Mehrotra, and N. Venkatasubramanian, "Energy efficient data collection in distributed sensor environments," in *Proc. of The Twenty Fourth IEEE Int'l Conf. on Distributed Computing Systems*, March 2004.

[24] J. Huang, W.-C. Feng, N. Bulusu, and W.-C. Feng, "Cascades: Scalable, flexible and composable middleware for multimodal sensor networking applications," in *Proc. of The ACM/SPIE Multimedia Computing and Networking (MMCN 2006)*, January 2006.

[25] R. R. Brooks, M. Zhu, J. Lamb, and S. S. Iyengar, "Aspect-oriented design of sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 853–865. Orlando, FL, USA: Academic Press, Inc., 2004.

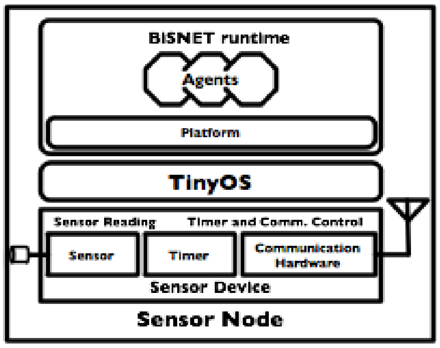

Figure 1. BiSNET Runtime Architecture

while $true$

do

- Read sensor data and convert the data to energy(E_F).
- Update energy level $(E(t))$
- **if** $E(t) <$ *death threshold* (T_D)
  - **then** Invoke the death behavior.
- **if** $E(t) >$ *replication pheromone emission threshold* (T_P)
  - **then** Emit a replication pheromone.
- **while** $E(t) >$ *replication threshold*(T_R) **and** *replication pheromone concentration*$(P_i) >$ *stimulation threshold* (T_{S_i})
  - **do**
    - Make a child agent.
    - Give the half of the current energy level to the child agent.
- Deposit energy units (E_P) to the local platform.
- **if** *the local platform is in the broadcast state*
  - **then**
    - **if** *# of agents in the local platform* > 1
      - **then**
        - Place a migration pheromone on the local node.
        - Move to a neighboring node.

Figure 2. Agent Actions in Each Duty Cycle

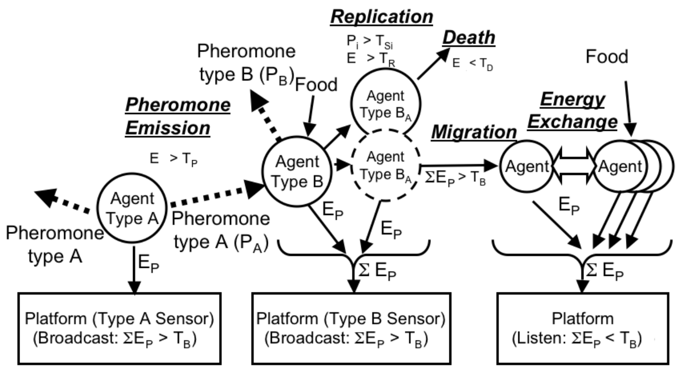

Figure 3. Agent Behaviors

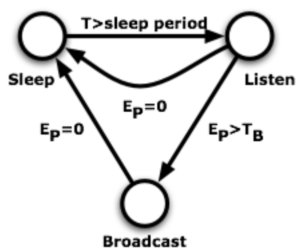

Figure 4. Platform State Transition

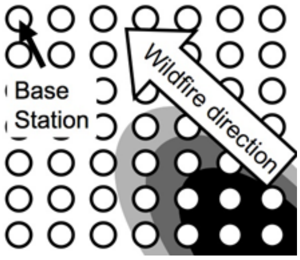

Figure 5. Simulated Network

Table 1. The Total Number of Collected and Reported Sensor Data

	# of collected data	# of reported data	Success rate
BiSNET-Mb	420	400	95.24%
BiSNET-RMb	485	460	94.85%
BiSNET-RMbm	488	462	94.67%
BiSNET-RMbr	488	462	94.67%
BiSNET-RMbmr	488	462	94.67%
GBR	380	360	94.74%
GBR-R	380	360	94.74%
GBR-P	380	358	94.21%
GBR-S	380	360	94.74%

Table 2. Average Latency of Data Transmission

	Latency (sec)	Standard Deviation
BiSNET-Mb	35	4.5
BiSNET-RMb	33	4
BiSNET-RMbm	34	4
BiSNET-RMbr	37	4
BiSNET-RMbmr	36	6
GBR	33	4
GBR-R	34	5
GBR-P	34	5
GBR-S	35	5

Table 3. Average Power Consumption of Each Sensor Node

	Power Consumption (mW)	Standard Deviation	Power Consumption per Reported Sensor Data (mW)
BiSNET-Mb	4158	1558	10.4
BiSNET-RMb	3924	1212	8.5
BiSNET-RMbm	4041	984.5	8.8
BiSNET-RMbr	4011	836	8.7
BiSNET-RMbmr	4044	745	8.6
GBR	3930	1015	10.9
GBR-R	3660	910	10.2
GBR-P	3480	850	9.7
GBR-S	3480	840	9.7

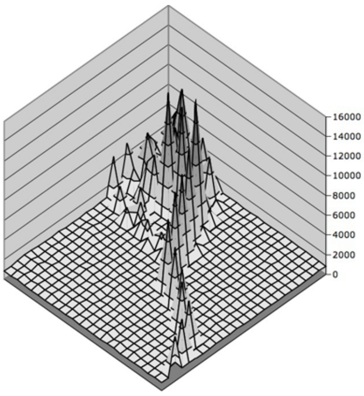

Figure 6. Power Consumption in BiSNET-Mb

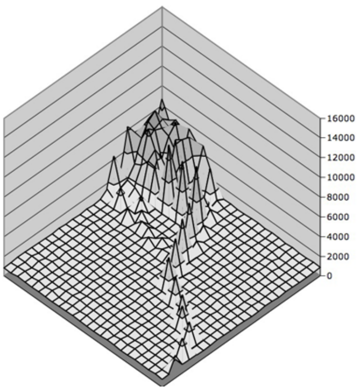

Figure 7. Power Consumption in BiSNET-RMb

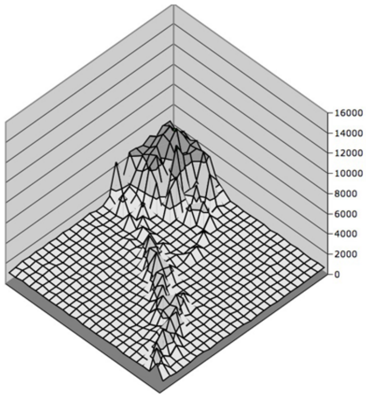

Figure 8. Power Consumption in BiSNET-RMbm

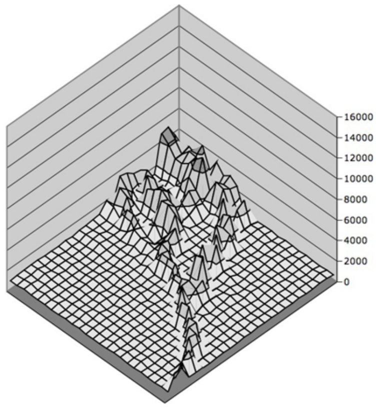

Figure 9. Power Consumption in BiSNET-RMbr

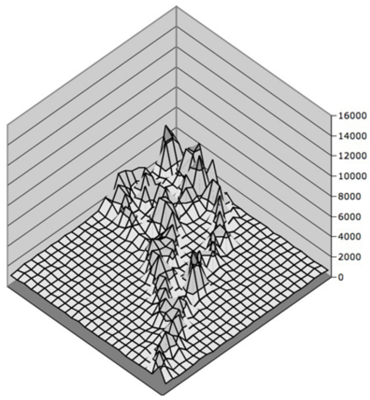

Figure 10. Power Consumption in BiSNET-RMbmr

Table 4. Memory Footprint in a MICA2 Mote

	ROM (KB)	RAM (KB)
BiSNET	1.0	24
Blink	0.04	1.6
GBR	0.84	26
Agilla	3.59	41.6

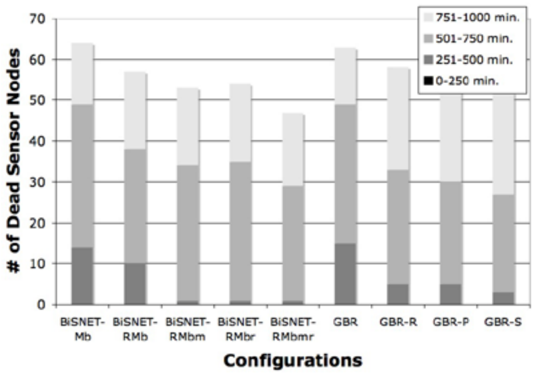

Figure 11. Network Lifetime