Biologically-Inspired Adaptive Data Aggregation for Multi-Modal 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 several key issues in multi-modal wireless sensor networks such as autonomy, adaptability, self-healing and simplicity. Based on the observation that various biological systems have developed mechanisms to overcome these issues, BiSNET implemenets certain biological mechanisms such as energy exchange, pheromone emission, replication, and migration to design sensor network applications. This paper presents the biologically-inspired mechanisms in BiSNET, and evaluates their impacts on the issues described above. Simulation results show that BiSNET allows sensor nodes to autonomously adapt their duty cycle intervals for power efficiency and responsiveness of data transmission, to adaptively aggregate data from different types of sensor nodes, to collectively self-heal (i.e., detect and eliminate) false positive sensor data, and to be lightweight.

1. Introduction

Multi-modal wireless sensor networks (MWSNs) face several challenges[1]. The first challenge is *autonomy*. Since sensor nodes can be deployed in an unattended area (e.g., forest and ocean) or physically unreachable area (e.g., inside a building wall), they are required to operate with the minimum aid from base stations or human administrators.

The second challenge is *adaptability*. Sensor nodes are required to adapt their operations to the environmental conditions that they monitor (e.g., temperature and carbon monoxide (CO)) [1, 2]. For example, sensor nodes may increase their duty cycle intervals when there is no significant change in their sensor readings. This results in less power consumption in the nodes. Also, when neighboring nodes report environmental changes, a sensor node may draw inference from the reports and decrease its duty cycle interval to be more watchful for a potential local environmental change in the future. This can increase responsiveness of the node to transmit its sensor data to a base station. In addition, a sensor node may aggregate data from different types of sensor nodes (e.g., temperature and CO data) and transmit the aggregated data to a base station. This can reduce power consumption in the nodes on the path toward the base station.

The third challenge is *self-healing*. Sensor reading usually contains some noises; it may be a false positive due to, for example, malfunction of sensors. Sensor nodes are required to self-heal (i.e., detect and eliminate) false positives in their sensor readings instead of transmitting them to base stations [3]. This can reduce power consumption of sensor nodes because in-sensor data processing incurs much less power consumption than data transmission does [4].

The fourth challenge is *simplicity*. Sensor control software needs to be simple in its design and small in its footprint because of limited availability of CPU power, memory and battery.

This paper describes an architecture for MWSNs, called BiSNET (Biologically-inspired architecture for Sensor NETworks), which addresses the above four challenges. BiSNET is motivated by the observation that various biological systems have already developed mechanisms to overcome those challenges [5, 6]. For example, bees act autonomously, influenced by local conditions and local interactions with other bees. A bee colony adapts to dynamic environmental conditions. 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 expand the hive. Also, bees recover (or self-heal) their pheromone traces to flowers when a part of them is lost. The structure and behavior of each bee are very simple; however, a group of bees autonomously exhibits desirable system characteristics such as adaptability and self-healing through collective behaviors and interactions among bees.

The BiSNET runtime operates atop of TinyOS in each sensor node (Figure 1). It consists of a middleware plat-

[1] An MWSN deploys multiple types of sensor nodes in an observation area (e.g., temperature, humidity and carbon monoxide (CO) sensors). Data from different types of sensor nodes are aggregated, through in-network processing or base station processing, to provide a multi-dimensional view of observed environmental conditions.

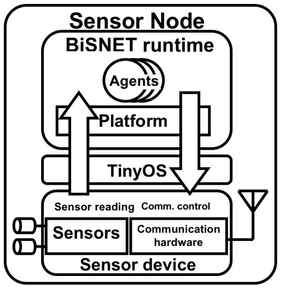

Figure 1. BiSNET Runtime Architecture

form 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/storage 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 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, adaptability, self-healing and simplicity of MWSNs. Simulation results show that BiSNET allows sensor nodes to autonomously adapt their duty cycle intervals for power efficiency, draw inference on potential environmental changes from sensing activities of neighboring nodes, adaptively aggregate data from different types of nodes, and collectively self-heal (i.e., detect and eliminate) false positive sensor data. The BiSNET runtime is lightweight thanks to a set of simple biological mechanisms.

2. Contributions

This section summarizes the contributions of this work.

- **Adaptive and decentralized duty cycle management:** BiSNET is the first attempt to investigate dynamic duty cycle management that adaptively balances the tradeoff between power efficiency and sensing responsiveness for potential environmental changes (i.e., the risk to miss significant environmental changes during sleep period). Each sensor node autonomously adjusts its duty cycle interval in a decentralized manner.
- **A simple and generic architectural design:** BiSNET applies a small number of simple biological concepts coherently for designing the mechanisms that address key challenges in MWSNs. It provides a simple and generic solution to design those multiple mechanisms (i.e., the mechanisms for adaptive data transmission, data aggregation, self-healing, power efficiency and inference) simultaneously, rather than focusing on a single mechanism. The simplicity of the biologically-inspired mechanisms in BiSNET allows the BiSNET runtime to be lightweight.

3. Design Principles for BiSNET Agents

(1) Decentralization: Similar to 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 [7, 8] and by avoiding any central coordination in deploying agents [9].

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

(3) Food gathering and storage: 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) in each duty cycle, and digest it to *energy* (honey)[2]. (Energy gain is proportional to an absolute change between the current and previous sensor data.) They keep some of the energy and deposit the rest in the local platform (hive).

(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 sensor node for reporting 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 an environment becomes more abundant, the population of agents dynamically changes based on their energy levels (i.e., changes in their sensor readings).

4. Design of BiSNET

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

4.1. 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

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

of a sensor node where the sensor data is captured. Agent designers can define arbitrary attributes for their agents.

Body implements the functionalities of the agent: food gathering and conversion (metabolism) of food to energy. Each agent gathers sensor data (as food) from the underlying sensor device and converts it to energy in each duty cycle. Depending on their agent types, different agents gather different types of sensor data.

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

- *Pheromone emission:* Agents may emit 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 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. Pheromones stimulate the agents on the local and neighboring nodes to replicate themselves.
- *Replication:* Agents may make a copy of themselves in response to the abundance of energy and pheromones. Each agent does not initiate replication until enough types of 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 pheromones. A child agent is placed on the platform 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 on a multi-hop and shortest-path basis.
- *Energy exchange:* Agents on a platform always share their energy units (honey) with each other so that their energy levels become equal. A migrating agent shares its energy units with other agents on a destination platform. Also, agents periodically deposit some of their energy units (honey) to their local platforms (hives).
- *Death:* Agents die due to lack of energy when they cannot balance energy gain and expenditure. The death behavior is intended to eliminate agents that carry false positive sensor data. When an agent dies, the underlying platform removes the agent and releases all resources allocated to the agent.

Every agent expends certain amount of energy to perform pheromone emission, replication and migration behaviors. The energy costs to invoke the behaviors are constant for all agents.

```
while $true$
do
    Read sensor data.
    Convert sensor data to energy $(E_F)$.
    Update energy level $(E(t))$ through energy exchange.
    if $E(t) <$ death threshold $(T_D)$
      then Invoke the death behavior.
    Reduce by half the concentration of each pheromone.
    if $E(t) >$ pheromone production threshold $(T_P)$
      then Emit a pheromone.
    while $E(t) >$ replication threshold $(T_R)$ and
    pheromone concentration$(P_i) >$ stimulation threshold $(T_{S_i})$
      do
        Make a child agent.
        Give the half of 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 Ask the local platform to move the agents to
          neighboring platforms.
        if pheromones exist on the local platform
          then Ask the local platform to move the pheromones to
          neighboring platforms.
```

Figure 2. Agent Actions in Each Duty Cycle

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.

$$E(t) = \frac{\sum_i^N E(t-1)}{N} + E_F \tag{2}$$

$E(t)$ 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 platform.

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)[3].

Then, an agent emits a pheromone if its energy level exceeds its pheromone emission threshold T_P (see Figures 2 and 3). Agents continuously adjust their 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 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 pheromone is emitted on a platform, all the agents on the platform can sense it. It may stimulate their replications. Each pheromone has its own concentration (or strength). It decays by half at each duty cycle. A pheromone completely evaporates (disappears) when its concentration becomes zero.

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 pheromones (P_i[4]) exceeds the pheromone's stimulation threshold T_{S_i} (see Figures 2 and 3). The agent keeps replicating itself until its energy level becomes less than its T_R. Agents continuously adjust their replication thresholds as the EWMA of their energy levels (Equation 4). The stimulation threshold of a 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 pheromone stimulation threshold for the 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.

Each agent deposits a certain amount of energy (E_P) to a 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 agents and pheromones to move to neighboring platforms (see also Figures 2 and 3)[5].

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 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 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 sensor network.

This adaptive data aggregation and transmission mechanism is designed with a self-healing capability in mind. When a sensor node does not work properly due to, for example, malfunctions, each agent on the node emits the pheromones that contain false positive sensor data. A large number of false positive pheromones may be transmitted to a neighboring node. However, they are discarded at the neighboring node 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 node. Also, agents stop emitting false positive pheromones on the malfunctioning node because their pheromone emission thresholds increase (see Equation 3).

4.2. 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 platform.

State controller dynamically changes the state of a sensor node to control its duty cycle. Each sensor node can be

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

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

[5] All agents migrate from a platform whose energy gain is greater than T_B, except a randomly selected agent. If there is only one agent in a platform, the agent cannot migrate. At least one agent runs on a platform.

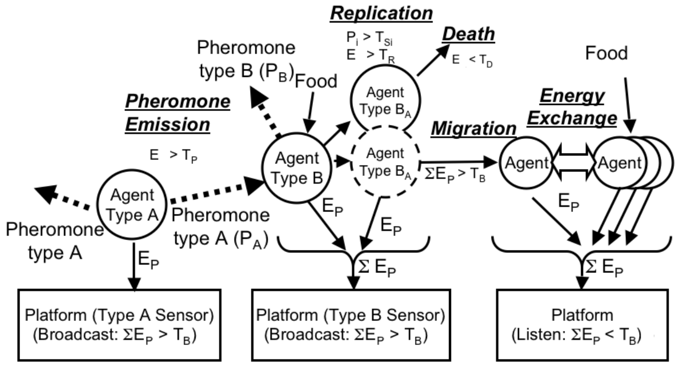

Figure 3. Agent Behaviors

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 either state, each agent performs a series of actions described in Figure 2.

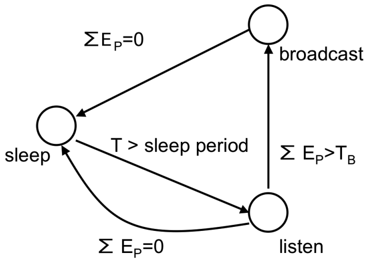

Figure 4. Platform State Transition

In the listen state, a platform turns on a radio receiver to receive data (agents and pheromones) from neighboring sensor nodes. 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 pheromones (the pheomone type i) available on the platform.

$$\text{sleep period} = \begin{cases} \frac{P_{\text{sleep}}}{\sum P_i} & \text{if } \sum P_i > 0 \\ P_{\text{sleep}} & \text{if } \sum P_i = 0 \end{cases} \qquad (7)$$

The sleep period is reverse proportional to the total concentration of pheromones available on a platform ($\sum P_i$). This means that a platform increases its duty cycle interval 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 pheromones from a neighboring node(s), it decreases its duty cycle interval even if there is no change in the sensor reading on the local node (see Equation 7). 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.

5. Simulation Results

This section shows simulation results to evaluate BiSNET in terms of adaptability, self-healing, inference, power efficiency and simplicity. BiSNET is implemented on TinyOS and evaluated in the TOSSIM simulator [10].

5.1. Application: Wildfire Detection

This simulation study emulates a sensor network deployed in a forest to detect wildfires. As shown in Figure **??**, this simulated network consists of 15 temperature sensors and 15 carbon monoxide (CO) sensors randomly deployed in a grid topology, and a wildfire moves from northeast to southwest. This paper focuses on Nodes 21 and 6. Node 21 detects a temperature change first, and then Node 6 detects the CO concentration changes next. Figure **??** shows how the two nodes sense environment changes over time. From the Figure, the temperature of node 21 changes from 80 Fahrenheit to reach the maximum at 240 Fahrenheit and then the temperature reduces to 80 Fahrenheit again. On the other hand, at Node 6, the CO concentration changes from 80 ppm (parts per million) to 240 ppm, which is in the range of the FAA's CO minimum performance standard for smoke detector[6], then the CO concentraion reduces to 80 ppm again.

5.2. Adaptive Data Transmission

Figure 5 shows the concentration of pheromones emitted by agents on Node 21 as well as the pheromone emission threshold on Node 21. The pheromone concentration increases when temperature spikes and drops, because agents emit more pheromones in response to higher energy intake. As the energy levels of agents grow, their pheromone emission thresholds increase as well (see Equation 3). This prevents agents to emit pheromones eventually. As Figure 5 shows, agents stop emitting pheromones when their pheromone emission threshold spikes (around 100th, and 160th minute), and pheromone concentration drops. Note that agents do not emit pheromones at all when there is no temperature changes. Agents adapt their pheromone emissions (i.e., sensor data transmission) to dynamic changes in their sensor readings.

[6] FAA's CO minimum performance standard is 200 ± 50 ppm

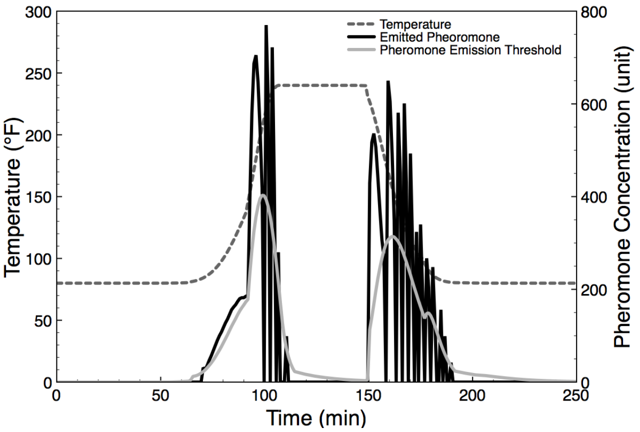

Figure 5. Pheromone Concentration and Pheromone Emission Threshold at Node 21

5.3. Inference

Figure 6 shows the pheromone concentrations on Node 6 and the number of replicating and migrating agents on Node 6. Temperature pheromones arrive Node 6 from Node 21 before CO level increases at Node 6. This allows the platform on Node 6 to draw inference from the temperature pheromones and reduce the node's sleep period (see Equation 7); therefore, the agents on Node 6 can start collecting more CO data before CO level increases, and the agents can start emitting CO pheromones immediately once CO level increases. This inference mechanism allows agents to be more responsive to environmental changes so that they can quickly replicate themselves and replicated agents can reach base stations in a shorter delay. In Figure 6, the responsiveness (the time lag between an environmental change and pheromone emission) on Node 6 is two times shorter than that on Node 21.

Figure 6 also shows that the agents on Node 6 perform replications only if the concentrations of both temperature and CO pheromones are high enough. As described in Section 4.1, agent replication (sensor data aggregation) is performed only when enough types of pheromones exhibit high concentrations.

Figure 7 depicts how sleep periods dynamically change on Nodes 21 and 6. On both nodes, platforms decreases the node's sleep periods when agents detect environmental changes, and increases it when agents detect no environmental changes. Platforms adapt their underlying nodes' sleep periods to environmental changes. Another finding from Figure 7 is that the sleep period of Node 6 decreases before CO level increases. This is because the platform on Node 6 receives temperature pheromones from Node 21 and the total pheromone concentration increases on Node 6, which in turn decreases the sleep period of Node 6. As

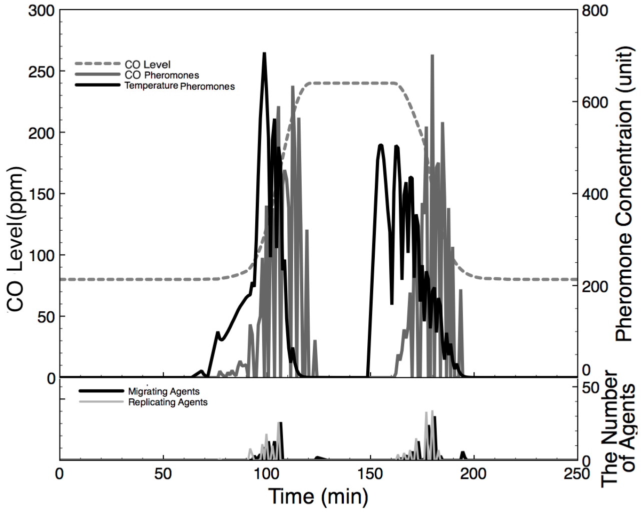

Figure 6. Pheromone Concentration and the Number of Replicating/Migrating Agents

described above, this inference mechanism increases the responsiveness of agents to environmental changes.

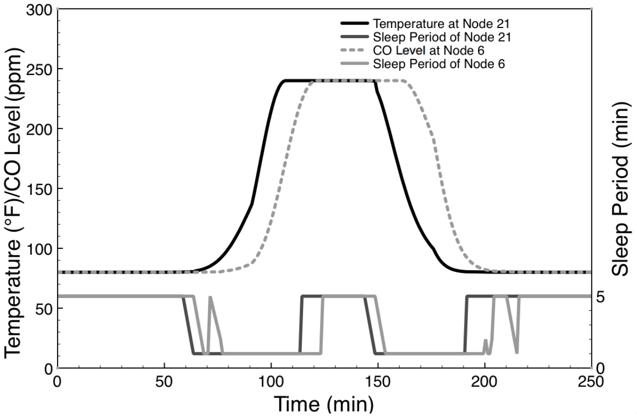

Figure 7. Sleep Period

Although the inference mechanism allows agents to collect more sensor data to be watchful for potential environmental changes, the they consume more power on sensor nodes. Table 1 summarizes this tradeoff. It shows the number of collected sensor data and power consumption at the node that performs inference (i.e., Node 6) and the node that does not perform it (i.e., Node 21). The data collection and power consumption are measured between when temperature/CO level spikes from 80 to 240 degrees/ppm and when it drops back to 80 degrees/ppm (for 200 minutes approximately). As shown in Table 1, by drawing inference from the pheromones emitted from Node 21, Node 6 collects 16.52% more data with only 3.55% more power consumption. The authors of the paper believe that extra power consumption is small enough to perform inference and BiSNET balances the tradeoff between sensing responsiveness and power consumption.

Table 1. Data Collection and Power Consumption of Node 6

	# of collected data	Power consumption
Without inference (Node 21)	230	3930 mA
With inference (Node 6)	268	3795 mA
Rate of increase	16.52%	3.55%

5.4. Power Efficiency through Adaptive Duty Cycle Management

Figure 8 shows the power consumption of Nodes 21 and 6. In the beginning of a simulation, the sensor nodes consume power to discover neighboring sensor nodes and set up network topology. After that, they minimize power consumption by increasing their sleep periods because there is no significant changes in their sensor readings. When temperature/CO level spikes, the power consumption of the sensor nodes spikes too because they immediately decrease their sleep periods (see also Figure 7). As shown in Figure 8, the adaptive duty cycle mechanism in BiSNET allows sensor nodes (agents/platforms) to effectively save their power consumption when there is no significant environmental change.

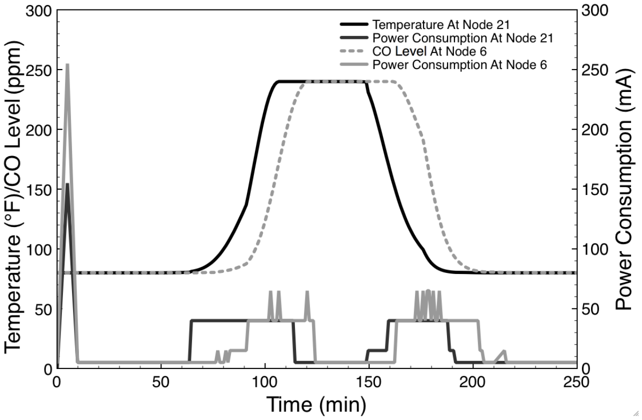

Figure 8. Power Consumption

Currently, platforms dynamically adjust their sleep periods between one and five minutes (see Figure 7). Table 2 compares the power consumption of sensor nodes (agents/platforms) with that of the configurations in which sleep period is fixed one or five minutes. Power consumption is measured between when temperature/CO level spikes from 80 to 240 degrees/ppm and when it drops back to 80 degrees/ppm (for 200 minutes approximately). Compared with the five minutes (fixed) duty cycle management, BiSNET consumes only 5% more power. BiSNET sacrifices the 5% power consumption to improve the sensing responsiveness against environmental changes. The fixed duty cycle management scheme cannot responsively sense and report environmental changes as BiSNET does. Compared with the one minute (fixed) duty cycle management, BiSNET consumes only 24% of power used by the fixed scheme. BiSNET effectively reduces power consumption by decreasing duty cycle only when necessary.

Table 2. Power Consumption in Different Configurations of Duty Cycle Management

Sleep period	Power consumption
1-5 mins (variable; BiSNET)	3930 mA
5 mins (fixed)	3740 mA
1 mins (fixed)	16340 mA

5.5. Power Efficiency through Data Aggregation

In addition to adaptive duty cycle management, pheromone-based data aggregation contributes to reduce power consumption of sensor nodes. Table 3 compares the power consumption of Node 6 in the two configurations that agents perform data aggregation and do not. When agents do not perform data aggregation, agents use only energy level to decide whether they replicate themselves. (They do not use pheromones.) Table 3 shows that Node 6 consumes power 4.9% less when agents perform data aggregation.

Table 3. Current Consumption with/without data aggregation

	Current consumption
with data aggregation	3930 mA
without data aggregation	4123 mA
Rate of increase	4.9%

5.6. Self-Healing of False Positive Data

Figures 9 and 10 demonstrate how each sensor node self-heals (i.e., detects and eliminates) false positive data when it or neighboring node malfunctions. BiSNET provides two self-healing capabilities: *intra-node* and *inter-node* self-healing. Figure 9 shows a result of intra-node self-healing. In this case, Node 21 is configured to malfunction and generate temperature data of 0 and 200 degrees repeatedly. When Node 21 starts malfunctioning, agents emit a large number of temperature pheromones very often because sensor data widely swings between 0 to 200 degrees. (The energy intake of agents is very high on Node 21.) However, the pheromone emission thresholds of agents rapidly grow as the agents' energy levels increase (see also Equation 3); within two minutes, agents start suppressing their pheromone emission. In five minutes, the concentration of pheromones dramatically drops, and no pheromones are emitted after five minutes. Accordingly, the pheromones are not transmitted to neighboring nodes in 6 minutes even if Node 21 keeps malfunctioning.

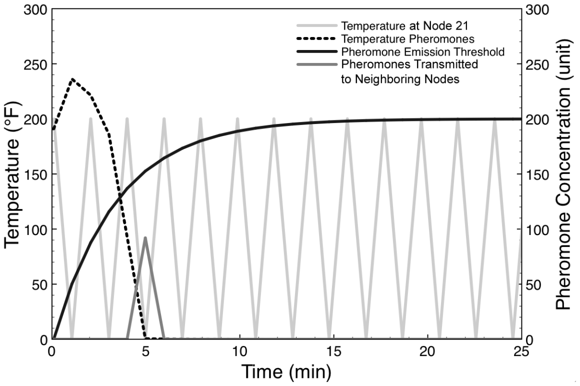

Figure 9. Intra-Node Self-Healing of False Positive Data

Figure 10 shows a result of inter-node self-healing. In this case, Node 21 works properly; however, Node 6 malfunctions. Node 6 periodically propagates a large number of CO pheromones and transmits them to Node 21. (Node 6 does not perform intra-node self-healing in this simulation scenario.) Since Node 21 does not detect temperature changes, the agents on Node 21 do not emit any temperature pheromones. As a result, the agents do not replicate themselves at all. Thus, even if Node 21 keeps accepting CO pheromones from Node 6, the agents on Node 21 totally ignore those pheromones. Using intra-node and inter-node self-healing, BiSNET allows sensor nodes (agents) to autonomously self-heal, i.e., detect and eliminate, false positive data (pheromones) and avoid wasting power.

5.7. Simplicity: Memory Footprint

In order to evaluate the simplicity of BiSNET, Tables 4 and 5 show the memory footprint of the BiSNET platform. Table 4 depicts the footprint in the PowerTOSSIM simulator on a PC, and compares it with the footprint of Blink (an example program in TinyOS), which periodically turns on and off an LED. Table 5 depicts the footprint in a MICA2 mote, and compares it with that of Blink and Agilla, which is a mobile agent platform for sensor networks [11]. As shown in these two tables, the BiSNET platform is fairly lightweight in its footprint, and it can be deployed on sensor devises whose resource availability is severely limited.

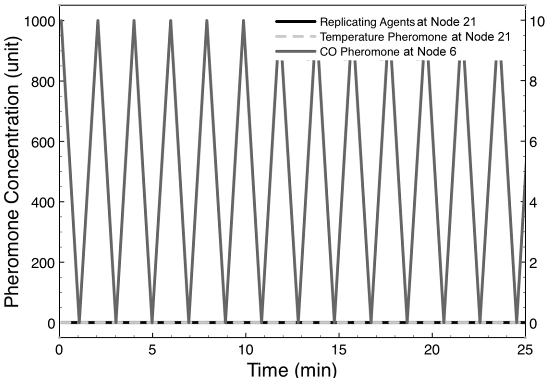

Figure 10. Inter-Node Self-Healing of False Positive Data

Table 4. Memory Footprint in PowerTOSSIM

	ROM (KB)	RAM (KB)
BiSNET	75	2406
Blink	34	648

Table 5. Memory Footprint in a MICA2 Mote

	ROM (KB)	RAM (KB)
BiSNET	0.7	18
Blink	0.04	1.6
Agilla	3.59	41.6

6. Related Work

In the previous work of the authors of the paper [12], BiSNET did not support MWSNs; it supported only one type of sensor nodes throughout a network. In this paper, BiSNET is extended to support MWSNs by introducing the concept of pheromones. Pheromones are used, in the agent replication behavior, for each agent to aggregate different types of sensor data and self-heal false positive data. Pheromones are also used for each platform to perform adaptive duty cycle management for power efficiency and inference on potential environmental changes for sensing responsiveness.

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, [13] 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, data aggregation and self-healing of false positive data.

[14] 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 [11]. 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 5, BiSNET is much more lightweight than Agilla.

[15, 16, 17] 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 [15, 16, 17] 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 [18]. 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.

SASHA proposes a self-healing mechanism by applying immunological mechanisms for base stations to identify fault sensor nodes [19]. A base station detects fault nodes by comparing data from multiple sensor nodes. In BiSNET, individual sensor nodes self-heal false positive sensor data in a decentralized manner. Since false positive data are not transmitted to base stations, BiSNET consumes less power for self-healing than SASHA. In fact, BiSNET does not incur any extra computing and communication overhead for self-healing. Self-healing is achieved as a result of agents to make decisions on whether they replicate themselves based on the concentration of pheromones.

[20] 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. [20] 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, [20] 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.

[21] 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. [21] 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 [21] has 133 states and 232 transitions among them. [21] does not consider design simplicity and runtime lightweightness. In addition, BiSNET operates in a decentralized manner, while [21] organizes a sensor network in a hierarchical manner.

7. Concluding Remarks

This paper describes a biologically-inspired sensor networking architecture, called BiSNET, which addresses several key issues in MWSNs such as autonomy, adaptability, self-healing and simplicity. This paper describes the biologically-inspired mechanisms in BiSNET and evaluates their impacts on the autonomy, adaptability, self-healing and simplicity of MWSNs. Simulation results show that

BiSNET allows sensor nodes to autonomously adapt their duty cycle intervals for battery efficiency, to draw inference on potential environmental changes from sensing activities of neighboring nodes, to collectively self-heal (i.e., detect and eliminate) false positive sensor data, to aggregate data from different types of nodes, and to be simple and lightweight.

Several extensions to BiSNET is planned. Currently, BiSNET assumes a traditional routing mechanism to transmit sensor data (i.e., agents) toward base stations. An biologically-inspired routing mechanism will be investigated to effectively direct agents to base stations.

References

[1] 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.

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

[3] P. Rentala, R. Musunuri, S. Gandham, and U. Sexena, "Survey on sensor networks," in *Proc of Int'l Conf. on Mobile Computing and Networking*, 2001.

[4] G. Mathur, P. Desnoyers, D. Genesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," in *Proc. of IEEE/ACM Conf. on Information Processing in Sensor Networks*, April 2006.

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

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

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

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

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

[10] V. Shnayder, M. Hempstead, B. rong Chen, G. Werner-Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *The Second ACM Conf. on Embedded Networked Sensor Systems (SenSys'04)*, November 2004.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] T. Bokareva, N. Bulusu, and S. Jha, "SASHA: Towards a self-healing hybrid sensor network architecture," in *Proc. of The Second IEEE Int'l Workshop on Embedded Networked Sensors*, May 2005.

[20] 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.

[21] 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.