ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Computer Networks xxx (2007) xxx–xxx

Computer Networks

www.elsevier.com/locate/comnet

BiSNET: A biologically-inspired middleware architecture for self-managing wireless sensor networks

Pruet Boonma *, Junichi Suzuki

Department of Computer Science, University of Massachusetts, Boston, MA 02125, United States

Abstract

This paper describes BiSNET (Biologically-inspired architecture for Sensor NETworks), a middleware architecture that addresses several key issues in multi-modal wireless sensor networks (MWSNs) such as autonomy, scalability, adaptability, self-healing and simplicity. Based on the observation that various biological systems have developed mechanisms to overcome these issues, BiSNET follows certain biological principles such as decentralization, food gathering/storage and natural selection to design MWSN applications. In BiSNET, each application consists of multiple software agents, which operate on the BiSNET middleware platform in individual sensor nodes, and each agent exploits certain biologically-inspired mechanisms such as energy exchange, pheromone emission, replication, migration and death. This is analogous to a bee colony (application) consisting of multiple bees (agents). This paper describes the biologically-inspired mechanisms in BiSNET, and evaluates their impacts on the autonomy, scalability, adaptability, self-healing and simplicity of MWSNs. Simulation results show that BiSNET allows sensor nodes (agents and platforms) to be scalable with respect to network size, autonomously adapt their sleep periods for power efficiency and responsiveness of data collection, adaptively aggregate data from different types of sensor nodes, and collectively self-heal (i.e., detect and eliminate) false positive sensor data. The BiSNET platform is implemented simple in its design and lightweight in its memory footprint.

Keywords: Self-managing wireless sensor networks; Middleware support for Self-managing sensor network applications

1. Introduction

This paper describes a middleware architecture for multi-modal wireless sensor networks (MWSNs),[1] called BiSNET (Biologically-inspired architecture for Sensor NETworks), which inherently addresses five challenges in MWSNs. 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 [1,2].

The second challenge is *scalability*. In order to cover large spatial extents or monitor the extents at a high-resolution, sensor networks are required

* Corresponding author.
E-mail addresses: pruet@cs.umb.edu (P. Boonma), jxs@cs.umb.edu (J. Suzuki).

[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, to provide a multi-dimensional view of collected sensor data.

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.06.006

to scale to a large number of sensor nodes[2] and a large amount of data generated by sensor nodes [1,3].

The third 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)) [3–6]. For example, sensor nodes may increase their duty cycle intervals (sleep periods) 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 (e.g., changes in temperature or CO level), a sensor node may draw inference from the reports and decrease its sleep period 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 a path toward the base station.

The fourth challenge is *self-healing*. Sensor reading usually contains some noise; 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 [5,7]. This can reduce power consumption of sensor nodes because in-node data processing consumes much less power than data transmission does [8].

The fifth challenge is *simplicity*. Sensor control software (e.g., applications and middleware) needs to be simple in its design and small in its footprint because of limited availability of CPU power, memory and battery.

In order to address the above five issues, BiSNET provides a middleware platform, called the BiSNET platform. The BiSNET platform hides low-level operating and networking details (e.g., network I/O and state control of sensor nodes) from applications, and implements a series of mechanisms to support autonomous, scalable, adaptive and self-healing applications. BiSNET also provides a high-level programming abstraction to aid the simple and rapid development of applications. The design of BiSNET is motivated by the observation

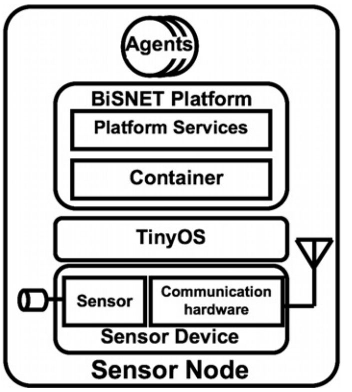

Fig. 1. BiSNET platform architecture.

that various biological systems have already developed mechanisms necessary to overcome those challenges [9,10]. 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. 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 is 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. Based on this observation, the authors of the paper believe that, if MWSN applications are designed after certain biological principles and mechanisms, they may be able to meet the requirements in MWSNs (i.e., autonomy, scalability, adaptability, self-healing and simplicity).

The BiSNET platform operates atop TinyOS in each sensor node to host applications (Fig. 1). In BiSNET, each application consists of multiple agents,[3] which follow several biological principles such as decentralization, autonomy, food gathering/storage and natural selection. This is analogous to a bee colony (application) consisting of multiple bees (agents) running on multiple platforms (hives). Each agent contains a set of data

[2] For example, the DARPA Networked Embedded Systems Technology program envisions networks consisting of 100 to 100000 simple computing nodes.

[3] Agents are software entities (software objects, components or modules); they do not represent any physical entity.

and program code, which is interpreted by the BiSNET platform at runtime. Each agent reads sensor data with the underlying sensor device, and discards or reports it to a base station using biological behaviors such as pheromone emission, replication and migration. The BiSNET platform consists of a container and platform services (Fig. 1). A container provides an execution environment for agents, and controls the state of the local sensor node (e.g., sleep, listen and broadcast). Platform services are used by agents 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, self-healing and simplicity of MWSN applications. Simulation results show that BiSNET allows sensor nodes (agents and platforms) to autonomously adapt their sleep periods 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 platform is lightweight thanks to a set of simple biologically-inspired 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 environmental changes (i.e., the risk to miss significant environmental changes during sleep period). The BiSNET platform allows each sensor node to autonomously adjusts its sleep period in a decentralized manner.
- *A simple and generic architectural design*: BiSNET applies a small number of simple biological concepts to design the mechanisms that address key challenges in MWSNs (e.g., the mechanisms for adaptive data transmission, data aggregation, self-healing, power efficiency and inference). Rather than implementing those mechanisms separately, BiSNET provides a simple and generic solution to implement the mechanisms simultaneously. The simplicity of the biologically-inspired mechanisms in BiSNET contributes to the simplicity and lightweightness of the BiSNET platform.

3. Design principles for BiSNET agents

In BiSNET, agents are designed after the following biological principles.

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 [11,12] and by avoiding any central coordination in deploying agents [13].
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).[4] (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 toward a base station on a hop-by-hop basis to report sensor data. An energy scarcity (an indication of few changes in sensor reading) eventually causes the death of agents. As in biological natural selection where more favorable species in an environment become more abundant, the population of agents dynamically changes based on their energy levels (i.e., changes in their sensor readings).

[4] 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 behavior.

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006

4. BiSNET

This section describes a programming abstraction for agents, the default agent the BiSNET platform provides, and the functions of the BiSNET platform.

4.1. BiSNET agent

The BiSNET platform provides a high-level programming abstraction for application developers to implement agents (i.e., MWSN applications) in an easy-to-understand manner. In BiSNET, each agent consists of *attributes*, *body* and *behaviors*. *Attributes* carry descriptive information on an agent. They include agent type (e.g., temperature sensing or 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 collected. Application developers can define arbitrary attributes for their agents.

Body implements the functionalities of an 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). Depending on their agent types, different agents collect 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. When an agent performs replication, it creates a new set of data and code (a new software agent). Each agent replicates itself only when enough types and concentration of pheromones become available on the local node. Individual pheromones are not independently transmitted to base stations. Certain types of high-concentration pheromones are grouped to stimulate agent replication. For example, an agent may replicate itself only when sufficient concentration of 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. When an agent perform migration from one sensor node to another, the BiSNET platform at a source node serializes the agent's data and code and transmits them to a destination node. The BiSNET platform on a destination node deserializes the transmitted data and code to run the agent.
- *Energy exchange*: Agents on each 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.

4.2. Default agent implementation

The BiSNET platform provides the default (or template) agent so that application developers can customize it to rapidly develop their own agents. Fig. 2 shows the body (a sequence of actions) that the default agent performs in each duty cycle. In this section, agent behaviors are visualized with the UML (Unified Modeling Language) sequence diagram.

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006

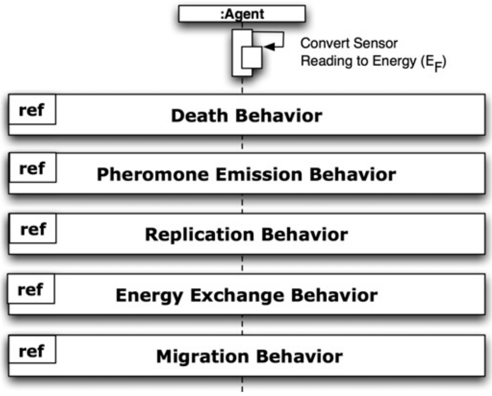

Fig. 2. Body of the default agent.

First, an agent reads sensor data (nectar) with the underlying sensor device, and converts it to energy (honey). The energy intake (E_F) is calculated with Eq. 1. S represents the absolute difference between sensor data in the current and previous duty cycles. M is the metabolic rate, which is a constant between 0 and 1.

$$E_{\mathrm{F}} = S \cdot M. \tag{1}$$

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. For example, if a MWSN is configured to be more sensitive to CO data than temperature data, the metabolic rate of CO sensor nodes should be greater than that of temperature sensor nodes.

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

$$E(t) = \frac{\sum_i^N E(t-1)}{N} + E_{\mathrm{F}}. \tag{2}$$

$E(t)$ 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 Figs. 2 and 3).[5]

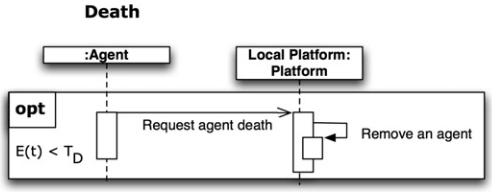

Fig. 3. Agent death behavior.

Then, an agent emits a pheromone if its energy level ($E(t)$) exceeds its pheromone emission threshold T_P (see Fig. 4). Agents continuously adjust their pheromone emission thresholds as the EWMA (exponentially weighted moving average) of their energy levels:

$$T_{\mathrm{P}}(t) = (1-\alpha)T_{\mathrm{P}}(t-1) + \alpha E(t), \tag{3}$$

$T_P(t)$ 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). The α value is a constant to control the sensitivity of T_P against the changes of E. A higher value of α changes T_P more sensitively against the recent changes in E. BiSNET uses a relatively low α value (0.25) in order to place more emphasis on the long-term transition trend of E. (Only significant changes in E have the effect of changing T_P.)

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[6]) exceeds the pheromone's stimulation threshold T_{S_i} (see Fig. 5). 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 (Eq. 4). The stimulation threshold of a pheromone changes as the EWMA of the pheromone's concentration (Eq. 5).

[5] 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.

[6] 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).

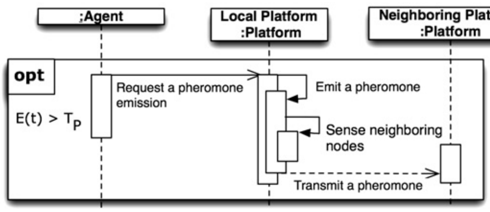

Fig. 4. Pheromone emission behavior.

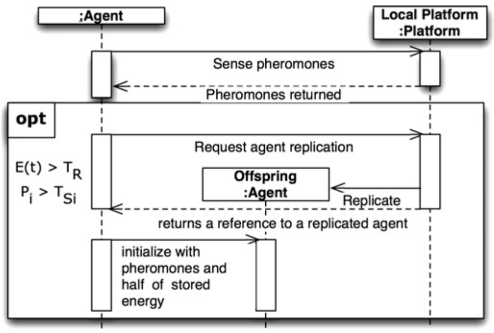

Fig. 5. Agent replication behavior.

$$T_{\mathrm{R}}(t) = (1-\beta)T_{\mathrm{R}}(t-1) + \beta E(t), \tag{4}$$

$$T_{\mathrm{S}_i}(t) = (1-\gamma)T_{\mathrm{S}_i}(t-1) + \gamma P_i(t). \tag{5}$$

$T_{\mathrm{R}}(t)$ is the current replication threshold, and $T_{\mathrm{R}}(t-1)$ is the one in the previous duty cycle. T_{S_i} is the current pheromone stimulation threshold for the pheromone type i, and $T_{\mathrm{S}_i}(t-1)$ 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. Similar to α, BiSNET uses a relatively low β and γ values (0.25) in order to place more emphasis on the long-term transition trend of E and P_i. (Only significant changes in E and P_i have the effects to change T_{R} and T_{S_i}, respectively.)

A replicating (parent) agent splits its energy units in to halves $\left(\frac{E(t)-E_{\mathrm{R}}}{2}\right)$, 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 the replication behavior.

Each agent deposits a certain amount of energy (E_{P}) to a platform that it resides on. Each agent strives to keep its energy level ($E(t)$) close to the one in the previous duty cycle ($E(t-1)$).

$$E_{\mathrm{P}} = \begin{cases} E(t) - E(t-1) & \text{if } E(t) \geqslant E(t-1) \\ 0 & \text{if } E(t) < E(t-1) \end{cases} \tag{6}$$

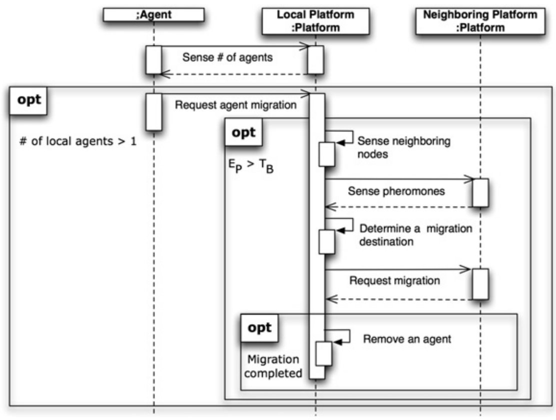

Fig. 6. Migration behavior sequence diagram.

When a platform's total energy gain ($\sum E_{\mathrm{P}}$) 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 Fig. 6).[7] In order for agents to determine which neighboring platforms they move to, each base station periodically propagates base station pheromones. (This is a different type of pheromones from those emitted by agents.) The concentration of base station pheromones 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.

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 extends 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 may emit the pheromones that contain

[7] 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 remains on a platform.

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. 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 (Eq. 3).

4.3. BiSNET platform

Each platform consists of *platform services* and a *container* (Fig. 1). *Platform services* hide lower-level computing and networking details, and provide high-level runtime services for agents to read sensor data and perform their behaviors. Example platform services are described below.

- *Agent behavior services* implement agent behaviors. Each of the agent behaviors described in Section 4.1 is implemented as a platform service. When agents perform their behaviors, they invoke corresponding platform services (Figs. 3–6). For example, when an agent replicates itself, it invokes the replication service that the local platform provides (Fig. 5).
- *Network I/O service* performs low-level wireless network communication such as data transmission and medium access control.
- *Localization service* implements a range-free area-based localization mechanism. This mechanism approximates the physical location of the local node by using base stations as reference points.[8] Upon receiving a base station pheromone from a base station, the localization service measures and records the topological distance (hop count) to the base station. Then, the service periodically transmits the distance information to the nearest base station. Base stations calculate the location of each node by performing multilateration with their locations.
- *Neighboring node maintenance service* keeps track of neighboring nodes. The service maintains a table on neighboring nodes, and updates the table when it receives agents, pheromones and base station pheromones from neighboring nodes. Agents use this service to emit pheromones to neighboring nodes and migrate toward base stations. It is also used by the localization service.
- *Pheromone management unit* maintains information about pheromone concentration at the local node and also in neighboring platforms.

A *container* maintains a reference table to the agents running on the local platform. Agents use the table to inspect which agents and how many agents are running on the local platform. Another responsibility of each container is to dynamically change the state of the underlying sensor node for controlling its duty cycle. Each sensor node can be in the *listen*, *broadcast* or *sleep* state (Fig. 7). 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 the series of actions described in Fig. 2.

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 Figs. 6 and 7). 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 (Fig. 7). The sleep period is determined as follows. P_{sleep} is a constant, and P_i is the concentration of each type of pheromones (the pheromone type i) available on the platform.

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

The sleep period is inversely proportional to the total concentration of pheromones available on a platform ($\sum P_i$). This means that a platform

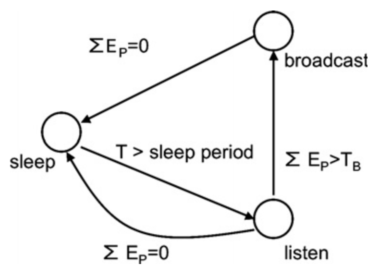

Fig. 7. Platform state transition.

[8] This paper assumes that each base station keeps track of its location with a GPS device and other nodes do not know their locations.

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 pheromones from a neighboring node(s) via the Pheromone Management Unit, it decreases its sleep period even if there is no change in the sensor reading on the local node (see Eq. 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 a sleep period.

5. Simulation results

This section shows a series of simulation results to evaluate BiSNET in terms of adaptability, scalability, self-healing, inference, power efficiency[9] and simplicity. BiSNET is implemented on TinyOS and evaluated in the PowerTOSSIM simulator [14].

5.1. Simulation configurations for a wildfire detection application

This simulation study emulates a sensor network deployed in a forest to detect wildfires. The simulated network consists of temperature sensors and CO sensors randomly deployed in a $N \times N$ grid topology (see Fig. 8). Two different size of networks are examined, 7×7 (49 nodes) and 25×25 (625 nodes). Half the sensor nodes equip temperature sensors, and the other half equip CO sensors. A wildfire moves from southeast to northwest, toward the base station located at the northwest corner of the sensor network. Simulations run with a fire spread model that describes wildfire spreading in nature [15].

This simulation study evaluates two different set of simulation results: *micro* evaluation and *macro* evaluation. The micro evaluation focuses on two sensor nodes in the network (nodes 21 and 6; see Fig. 8), and evaluates how BiSNET works across the two nodes. Node 21 detects a temperature change first, and then node 6 detects a CO level change next (Fig. 8). At node 21, temperature changes from 80°F to 240°F, and goes back to 80°. At node 6, CO level changes from 80 to 240 ppm (parts per million),[10] and goes back to 80 ppm.

The macro evaluation evaluates how BiSNET impacts the performance of the whole sensor network, such as the scalability against network size, success rate of sensor data transmission and network life.

5.2. Micro evaluation

This section presents micro evaluation results. Every micro evaluation is conducted with a 7×7 sensor network.

5.2.1. Adaptive data transmission

Fig. 9 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 Eq. 3). This prevents agents from emitting pheromones. As Fig. 9 shows, agents stop emitting pheromones when their pheromone emission threshold spikes (around 100th, and 160th min), and pheromone concentration drops. Note that agents do not emit pheromones at all when there is no temperature change. Agents adapt their pheromone emissions (i.e., sensor data transmission) to dynamic changes in their sensor readings.

5.2.2. Inference

Fig. 10 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 Eq. 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 the 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

[9] Voltage supply in sensor nodes is assumed to be constant; therefore, current in milliamperes is used as an indication of power.

[10] 240 ppm is in the range of the FAA CO minimum performance standard for smoke detectors (200 ± 50 ppm).

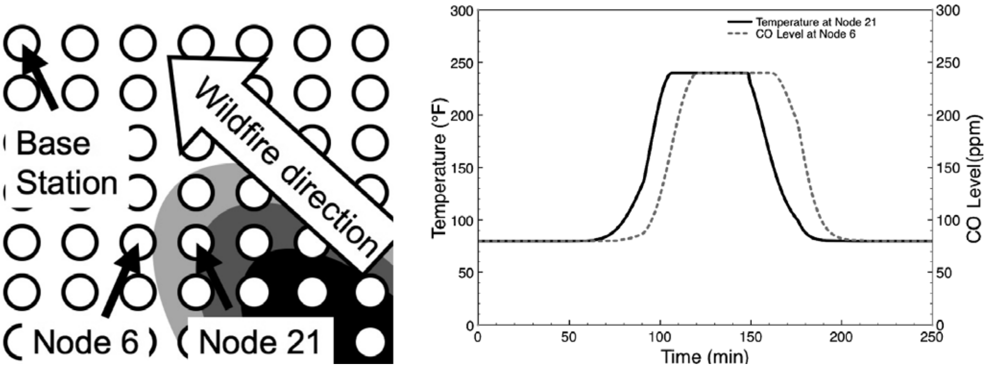

Fig. 8. Simulation configurations.

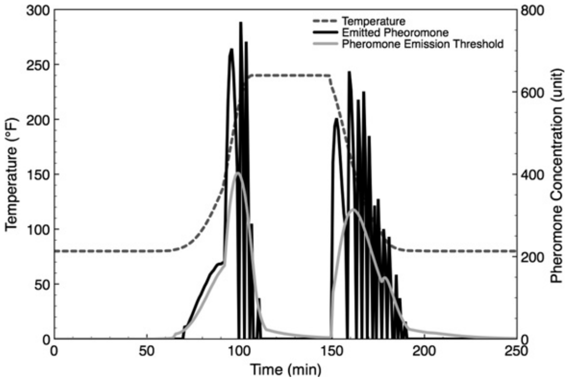

Fig. 9. Pheromone concentration and pheromone emission threshold at node 21.

stations with a shorter delay. In Fig. 10, the responsiveness (the time lag between an environmental change and pheromone emission) on node 6 is two times shorter than that on node 21.

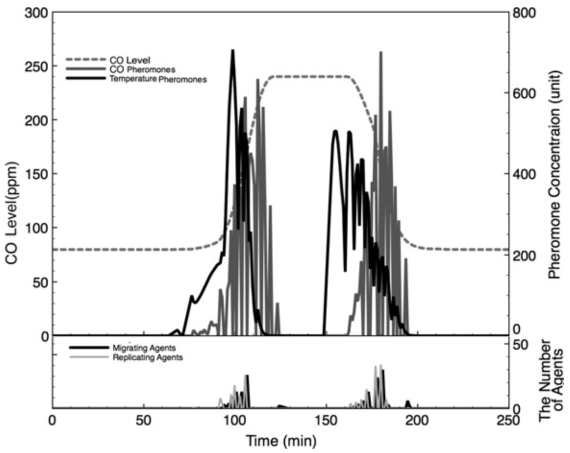

Fig. 10. Pheromone concentration and the number of replicating/migrating agents.

Fig. 10 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 4.2, agent replication (sensor data aggregation) is performed only when enough types of pheromones exhibit high-concentrations.

Fig. 11 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 Fig. 11 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 described above, this inference mechanism increases the responsiveness of agents to environmental changes.

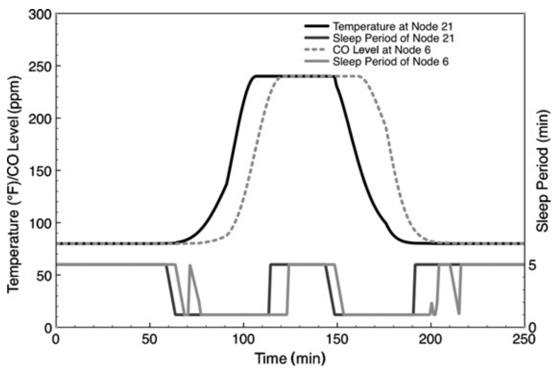

Fig. 11. Sleep period.

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 2 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°/ppm and when it drops back to 80°/ppm (for 250 min approximately). As shown in Table 2, 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.

5.2.3. Power efficiency through adaptive duty cycle management

Fig. 12 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 change in their sensor readings. When the temperature/CO level spikes, the power consumption of the sensor nodes spikes too because they immediately decrease their sleep periods (see also Fig. 11). As shown in Fig. 12, the adaptive duty cycle mechanism in BiSNET allows sensor nodes (agents/platforms) to effectively reduce their power consumption when there is no significant environmental change.

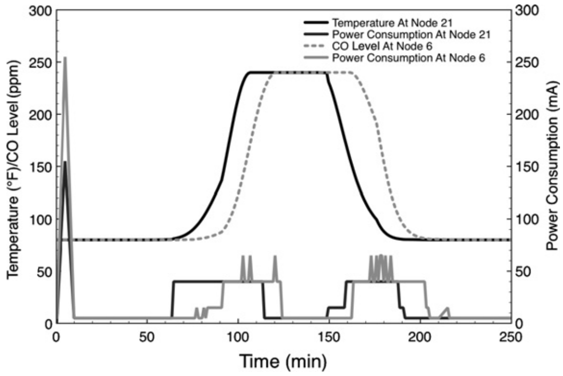

Fig. 12. Power consumption.

Currently, platforms dynamically adjust their sleep periods between 1 min and 5 min (see Fig. 11). Table 3 compares the number of collected data and the power consumption of sensor nodes (agents/platforms) with that of the configurations in which sleep period is fixed at 1 min or 5 min. The data collection and power consumption are measured between when temperature/CO level spikes from 80 to 240°/ppm and when it drops back to 80°/ppm (for 250 min approximately). Compared with the 5 min (fixed) duty cycle management, BiSNET consumes only 5% more power while collecting 16.52% more data. BiSNET sacrifices the 5% power consumption to improve the sensing responsiveness against environmental changes. A fixed duty cycle management scheme cannot responsively sense and report environmental changes as BiSNET does. Compared with the 1 min (fixed) duty cycle management, BiSNET consumes only 62% of the

Table 1
Simulation parameters

Parameters	Values
The number of sensor nodes	49 or 626
Metabolic rate (M)	1
EWMA coefficients (α, β and γ)	0.25
Agent death threshold (T_D)	1
Platform broadcast threshold (T_B)	25
Power consumption in the listen state	10 mA
Power consumption in the broadcast state	25 mA
Power consumption in the sleep state	5 mA
Maximum sleep period	5 min
Minimum sleep period	1 min

Table 2
Data collection and power consumption with and without inference

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

Table 3
Data collection and power consumption in different configurations of duty cycle management

Sleep period (in min)	Number of collected data	Power consumption (mA)
1–5 (variable; BiSNET)	268	3930
5 (fixed)	230	3740
1 (fixed)	268	6340

power used by the fixed scheme while collecting the same amount of data. In particular, the power consumption in listen state is much lower than in broadcast state (see Table 1), so even if the sleep period of a sensor node is shorten (i.e., sensor node spends more time in listen state), it does not affect the overall power consumption of a sensor node much. Moreover, from Fig. 11, the interval when the sleep period of node 6 (with inference, dynamically adjusted sleep period between 1 min and 5 min) does not overlap with that of node 21 (without inference, the sleep period is fixed at 5 min) is relatively small compared with with the whole simulation time; consequently, this contributes to a small increase in power consumption. A sensor node in listen state consumes much less batter power than in broadcast state; however, it is two times larger than that of sleep state. Therefore, a sensor node with 1 min fixed duty cycle has to waste battery power when there is no event (e.g., from 0 min to 55 min). BiSNET effectively reduces power consumption by decreasing duty cycle only when necessary.

5.2.4. 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 4 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 4 shows that node 6 consumes 4.9% less power when agents perform data aggregation.

5.2.5. Self-healing of false positive data

Figs. 13 and 14 demonstrate how each sensor node self-heals (i.e., detects and eliminates) false positive data when it or a neighboring node malfunctions. BiSNET provides two self-healing capabilities: *intra-node* and *inter-node* self-healing. Fig. 13 shows a result of intra-node self-healing.

Table 4
Power consumption with and without data aggregation

	Power consumption
Without data aggregation	4123 mA
With data aggregation	3930 mA
Reduction rate	4.9%

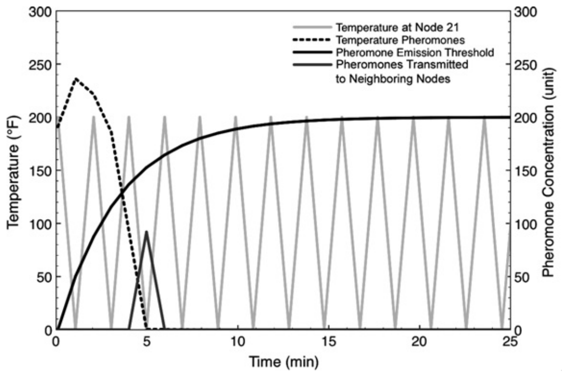

Fig. 13. Intra-node self-healing of false positive data.

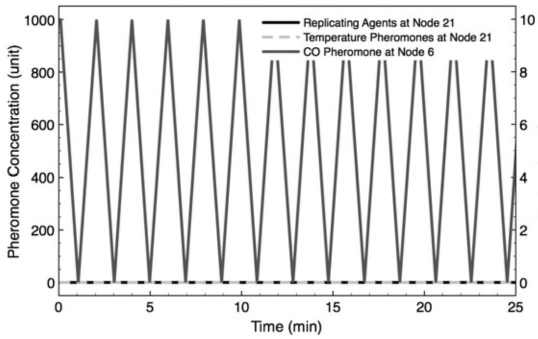

Fig. 14. Inter-node self-healing of false positive data.

In this case, node 21 is configured to malfunction and generate temperature data of 0 and 200° repeatedly. When node 21 starts malfunctioning, agents emit a large number of temperature pheromones very often because sensor data widely swings between 0° and 200°. (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 Eq. 3); within 2 min, agents start suppressing their pheromone emission. In 5 min, the concentration of pheromones dramatically drops, and no pheromones are emitted after 5 min. Accordingly, the pheromones are not transmitted to neighboring nodes in 6 min even if node 21 keeps malfunctioning.

Fig. 14 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.

5.2.6. Simplicity: memory footprint

In order to evaluate the simplicity of BiSNET, Table 5 shows the memory footprint of the BiSNET platform, a MICA2 mote. It also shows the footprint of Blink (an example program in TinyOS), which periodically turns on and off an LED, and a mobile agent platform for sensor networks called Agilla [16]. As shown in Table 5, the BiSNET platform is fairly lightweight in its footprint, and it can be deployed on sensor devices whose resource availability is severely limited.

5.3. Macro-level evaluation

This section presents macro evaluation results. Macro evaluation is conducted with a 7×7 and 25×25 sensor networks.

5.3.1. Impacts of simulation parameters on BiSNET performance

As Table 1 shows, this simulation study uses 0.25 for the value of α, β and γ in Eqs. (3)–(5). These parameters control the sensitivity for agents to emit pheromones, replicate themselves and migrate. This section discusses how different α, β and γ values impact the total power consumption of a sensor network (i.e., power consumption by all the sensor nodes in a network) and the number of data reported to the bases station throughout a simulation. This evaluation is conducted with a 7×7 sensor network.

Table 6 shows the power consumption of a sensor network and the number of reported data when α varies from 0.1 to 0.75. (β and γ are 0.25). When α increases, the number of reported data decreases because the pheromone emission threshold (T_{P}) becomes more sensitive to the changes in energy level; it is less likely that agents produce pheromones (see Eq. 3 and Fig. 4). As a result, agents replicate themselves less often, and less data is reported to the base station. On the other hand, when α is too small, agents produce pheromones too often and consume more power.

Table 7 shows the power consumption of a sensor network and the number of reported data when β varies from 0.1 to 0.75. (α and γ are 0.25). When β increases, the number of reported data decreases because the replication threshold (T_{R}) becomes more sensitive to the changes in energy level; it is less likely that agents replicate themselves (see Eq. 4 and Fig. 5). As a result, agents report less data to the base station. On the other hand, when the value of β is too small, agents replicate themselves too often, transmit more data and consume more power.

Table 8 shows the power consumption of a sensor network and the number of reported data when γ varies from 0.1 to 0.75. (α and β are 0.25). When γ increases, the number of reported data decreases

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

Table 6
Power consumption and the number of reported data with different α values

α	Power consumption (mA)	Number of reported data
0.1	4105	548
0.25	3614.08	548
0.5	3213	532
0.75	2880	518

Table 7
Power consumption and the number of reported data with different β values

β	Power consumption (mA)	Number of reported data
0.1	3912.3	548
0.25	3614.08	548
0.5	3113.5	488
0.75	2530.3	442

Table 8
Power consumption and the number of reported data with different γ values

γ	Power consumption (mA)	Number of collected data
0.1	3892	548
0.25	3614.08	548
0.5	3105.3	505
0.75	2588.5	482

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006

because the pheromone stimulation threshold (T_{S_i}) becomes more sensitive to the changes in pheromone concentration; it is less likely that agents replicate themselves (see Eq. 5 and Fig. 5). As a result, agents report less data to the base station. On the other hand, when γ is too small, agents replicate themselves too often, transmit more data and consume more power.

Currently, BiSNET uses a relatively low-value (0.25) for α, β and γ in order for agents to place more emphasis on the long-term transition of energy level and pheromone concentration while maintaining the sensitivity to them.

5.3.2. Data aggregation

Table 9 shows the total number of sensor data that all sensor nodes collect and transmit to the base station throughout a simulation in two different configurations with pheromones enabled and disabled. With pheromones enabled, sensor nodes collect and transmit more sensor data to the base station by taking advantage of the inference mechanism in BiSNET. The success rate of data transmission from sensor nodes to the base station decreases 1.33% with pheromones enabled. When a pheromone is emitted, it is propagated to neighboring nodes. This propagation can block agents from migrating toward the base station. (There is no data propagation with pheromones disabled.) However, the authors of the paper believe this reduction rate is acceptable enough.

Table 10 shows the average power consumption of each sensor node in two different configurations with pheromones enabled and disabled. With pheromones enabled, agents reduce the number of sensor data transmitted to the base station by aggregating sensor data. This contributes to 37.72% reduction of power consumption on each node, compared with the configuration with pheromones disabled. Moreover, the standard deviation of power consumption among nodes is smaller when using pheromones.

Table 9
The number of collected and reported sensor data with and without pheromones

	Number of collected data	Number of reported data	Success rate
Without pheromones	494	490	99.19%
With pheromones	560	548	97.86%
Change rate	13.36%	11.84%	1.33%

Table 10
Average power consumption of each sensor node

	Average power consumption (mA)	Standard deviation
Without pheromones	5371.84	4318.62
With pheromones	3614.08	1995.354
Reduction rate	32.72%	

Figs. 15 and 16 show how much power individual sensor nodes consume. (Each intersection of lines in a figure represents a sensor node.) As shown in Fig. 16, with pheromones disabled, some nodes (particularly, the nodes close from the base station) consumes much more power than others. This increases a risk that a network separates into islands and some nodes cannot communicate with the base station. In contrast, with pheromones enabled, every node consumes a lower and similar amount of power.

Fig. 17 shows the distribution of the number of sensor nodes against power consumption throughout a simulation. This figure suggests that the number sensor nodes is normally distributed against power consumption. As Table 10 shows as well, power consumption is better distributed over sensor nodes when using pheromones. This avoids network separations and contributes to extend network life.

5.3.3. Network lifetime

Fig. 18 shows how soon sensor nodes go down due to lack of power with pheromones enabled and disabled. The same battery capacity is assigned to both cases (750 mA h). With pheromones disabled, 15 of 49 nodes go down in the first

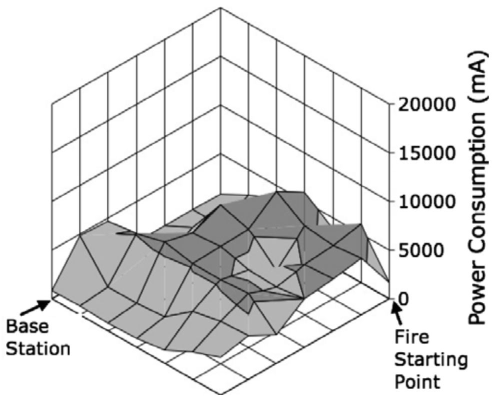

Fig. 15. Power consumption of sensor nodes with pheromones enabled.

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006

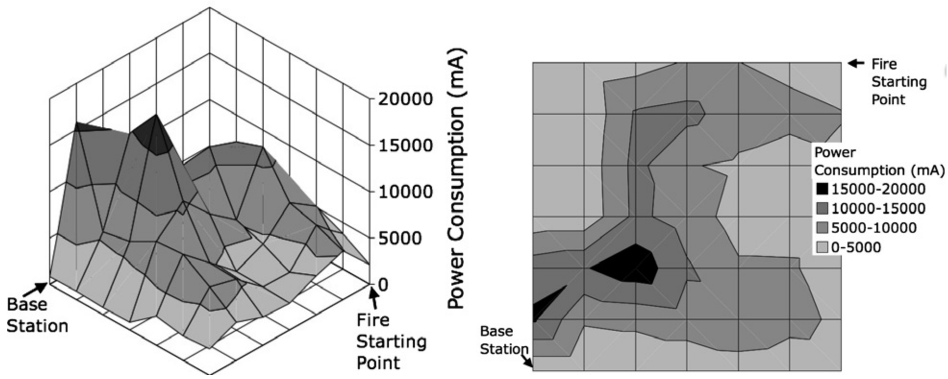

Fig. 16. Power consumption of sensor nodes with pheromones disabled.

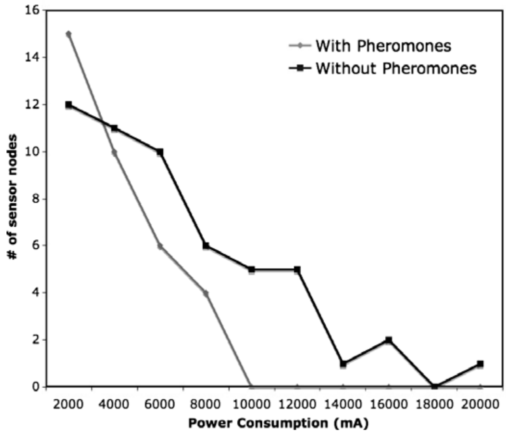

Fig. 17. Distribution of the number of nodes against power consumption.

150 min, and all nodes go down in 238 min. With pheromones enabled, 26 nodes go down in 250 min, and all nodes go down in 379 min. Using pheromones contributes to extend the network lifetime.

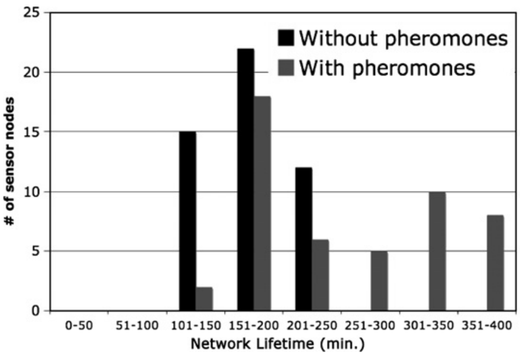

Fig. 18. Histogram of sensor node lifetime.

Table 11
The total number of collected and reported sensor data in a 25×25 network

	Number of collected data	Number of reported data	Success rate
Without pheromones	670	665	99.25%
With pheromones	774	762	98.45%
Change rate	15.52%	14.59%	0.81%

Table 12
Average power consumption of each sensor node in a 25×25 network

	Average power consumption (mA)	Standard deviation
Without pheromones	5542.4	4148.0
With pheromones	4024.104	1674.04
Reduction rate	27.39%	

5.3.4. Scalability

In order to evaluate the scalability of BiSNET against the number of nodes, a set of simulations was carried out with a larger sensor network consisting of 600 nodes. Table 11 shows the total number of sensor data that all sensor nodes collect and transmit to the base station throughout a simulation. Table 12 shows the average power consumption of sensor nodes throughout a simulation. These results are very similar to the results obtained from smaller-scale simulations with 49 nodes (see Tables 9 and 10). Simulation results show that BiSNET is scalable against the increase of network size and data volume generated by nodes.

6. Related work

This paper describes the research findings extending several previous works [17,18]. In [17], 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. In [18], BiSNET was examined through micro evaluation in a smaller network of 30 nodes. This paper includes additional micro evaluation results as well as macro evaluation results in larger networks of 49 and 625 nodes.

There are several research efforts applying biological mechanisms to sensor networks. For example, in order to synchronize clocks of sensor nodes in a decentralized manner, [19] 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.

Britton et al. [20] 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.

Szumel and Owens [21] proposes a generic communication primitive for sensor networks. The primitive hides lower-level implementation, i.e., network communication, while maintain the ability of programmer to control over the communication behavior of sensor node. The authors use a biological communication mechanism, called pheromone, as the communication primitive. A set of pheromone properties, i.e., type, strength, source, and payload, and instructions, i.e., deposit and smell is provided to the application developer to be used as the communication mechanism between sensor node. Different from BiSNET which is designed based on biological systems from the bottom up, the pheromone concept in this work is not properly integrated into the other part of sensor software development. Hence, application developers have to deal with two different levels of concept, a high-level concept of pheromone, and low-level concept of sensor node programming. 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 the 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 themselves with those issues when using BiSNET.

Agilla proposes a programming language to implement mobile agents for sensor networks, and provides a runtime system (interpreter) to operate agents on TinyOS [16]. 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 Table 5, BiSNET is much more lightweight than Agilla.

Hsin and Liu [22], Mišić and Mišić [23], Ye et al. [24] describe dynamic duty cycle management in sensor nodes. Their goal is to improve power efficiency, and they do not consider sensing responsiveness to 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 [22–24] randomly change sleep period or use other metrics such as the average time for a sensor node to process packets.

BiSNET is similar to [25] in that it proposes a mechanism that allows users to manually specify the sleep period of sensor nodes in order to balance the tradeoff between power efficiency and sensitivity of event detection. Unlike this mechanism, BiSNET allows sensor nodes to autonomously adjust their sleep periods without any aid from human users. Moreover, in BiSNET, different sensor nodes can have different sleep periods, depending on their sensor readings. This balances the tradeoff between power efficiency and sensitivity of event detection more effectively.

Quasar proposes a data collection protocol that balances the tradeoff between data accuracy and power efficiency [26]. 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 a 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 faulty sensor nodes [27]. 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 agent making decisions on whether they replicate themselves based on the concentration of pheromones.

Huang et al. [28] 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, so as not to sacrifice its performance and runtime lightweightness. Huang et al. [28] 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 Fig. 2 and Eqs. 4 and 5). In addition, [28] 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.

Brooks et al. [29] 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. Brooks et al. [29] is similar to BiSNET in that it can detect and eliminate false positive data as a function of collaborative sensing. However, BiSNET is designed much simpler; it has only a few states and transitions among them, while [29] has 133 states and 232 transitions among them. Brooks et al. [29] does not consider design simplicity and runtime lightweightness. In addition, BiSNET operates in a decentralized manner, while [29] 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, scalability, adaptability, self-healing and simplicity. Inspired from biological systems, BiSNET provides a set of simple yet generic solutions that address those issues simultaneously rather than focusing on them one by one or in an ad-hoc manner. This paper describes the biologically-inspired mechanisms in BiSNET and evaluates their impacts on these issues in MWSNs. Simulation results show that BiSNET allows sensor nodes (agents and platforms) to scale to network size, autonomously adapt their sleep periods for power efficiency, draw inference on potential environmental changes from sensing activities of neighboring nodes, collectively self-heal (i.e., detect and eliminate) false positive sensor data, and aggregate data from different types of nodes. The BiSNET platform is implemented so as to be simple in its design and lightweight in its memory footprint.

Several extensions to BiSNET are planned. Currently, BiSNET assumes a traditional decentralized routing mechanism to transmit sensor data (i.e., agents) toward base stations via shortest paths. A biologically-inspired routing mechanism will be investigated to effectively direct agents to base stations. In addition, BiSNET currently detects and eliminates false positive data. It is planned to investigate an additional mechanism to detect false negative data as well.

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006

References

[1] M. Yu, H. Mokhtar, M. Merabti, A survey of network management architecture in wireless sensor network, in: Proceedings of the Sixth Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting, 2006.

[2] J. Blumenthal, M. Handy, F. Golatowski, M. Haase, D. Timmermann, Wireless sensor networks – new challenges in software engineering, in: Proceedings of the Emerging Technologies and Factory Automation, 2003.

[3] S. Hadim, N. Mohamed, Middleware challenges and approaches for wireless sensor networks, IEEE Distributed Systems Online 7 (3) (2006) 853–865.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, Elsevier Journal of Computer Networks 38 (2002) 393–422.

[5] J. Ledlie, J. Schneidman, M. Welsh, M. Roussopoulos, M. Seltzer, Open problems in data collection networks, in: Proceedings of European SIGOPS, 2004.

[6] K. Romer, O. Kasten, F. Mattern, Middleware challenges for wireless sensor networks, ACM SIGMOBILE Mobile Computing and Communications Review 6 (4) (2002) 59–61.

[7] P. Rentala, R. Musunuri, S. Gandham, U. Sexena, Survey on sensor networks, in: Proceedings of International Conference on Mobile Computing and Networking, 2001.

[8] G. Mathur, P. Desnoyers, D. Genesan, P. Shenoy, Ultra-low power data storage for sensor networks, in: Proceedings of IEEE/ACM Conference on Information Processing in Sensor Networks, 2006.

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

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

[11] R. Albert, H. Jeong, A. Barabasi, Error and attack tolerance of complex networks, Nature 406.

[12] N. Minar, K.H. Kramer, P. Maes, Cooperating mobile agents for dynamic network routing, Software Agents for Future Communications Systems.

[13] G. Cabri, L. Leonardi, F. Zambonelli, Mobile-agent coordination models for internet applications, IEEE Computer.

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

[15] B. Drossel, F. Schwabl, Self-organized critical forest-fire model, Physical Review Letters 69 (11) (1992) 1629–1632.

[16] C.-L. Fok, G.-C. Roman, C. Lu, Rapid development and flexible deployment of adaptive wireless sensor network applications, in: Proceedings of International Conference on Distributed Computing Systems, 2005.

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

[18] P. Boonma, J. Suzuki, Biologically-inspired data aggregation for multi-modal wireless sensor networks, in: Proceedings of The 31th IEEE Conference on Local Computer Network (LCN), 2006.

[19] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, R. Nagpal, Firefly-inspired sensor network synchronicity with realistic radio effects, in: Proceedings of The Third International Conference on Embedded Networked Sensor Systems, 2005.

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

[21] L. Szumel, J.D. Owens, The virtual pheromone communication primitive, in: P.B. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), Proceedings of the Second IEEE International Conference on Distributed Computing in Sensor Systems, Lecture Notes in Computer Science, vol. 4026, Springer, 2006, pp. 135–149.

[22] C.-F. Hsin, M. Liu, Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms, in: Proceedings of The Third International Symposium on Information Processing in Sensor Networks, 2004.

[23] J. Mišić, V.B. Mišić, Duty cycle management in sensor networks based on 802.15.4 beacon enabled MAC, Ad Hoc and Sensor Wireless Networks Journal 1 (3) (2005) 207–233.

[24] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactioncs on Networking 12 (3) (2004) 493–506.

[25] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, Energy-efficient surveillance system using wireless sensor networks, in: Proceedings of the Second International Conference on Mobile System, Applications, and Services, 2004.

[26] Q. Han, S. Mehrotra, N. Venkatasubramanian, Energy efficient data collection in distributed sensor environments, in: Proceedings of The Twenty Fourth IEEE Internationl Conference on Distributed Computing Systems, 2004.

[27] T. Bokareva, N. Bulusu, S. Jha, SASHA: Towards a self-healing hybrid sensor network architecture, in: Proceedings of The Second IEEE International Workshop on Embedded Networked Sensors, 2005.

[28] J. Huang, W.-C. Feng, N. Bulusu, W.-C. Feng, Cascades: Scalable, flexible and composable middleware for multi-modal sensor networking applications, in: Proceedings of The ACM/SPIE Multimedia Computing and Networking, 2006.

[29] R.R. Brooks, M. Zhu, J. Lamb, S.S. Iyengar, Aspect-oriented design of sensor networks, Journal of Parallel and Distributed Computing 64 (7) (2004) 853–865.

Pruet Boonma is a second year Ph.D. student at the University of Massachusetts, Boston. His research interests include adaptive distributed systems and biologically-inspired wireless sensor networks. Mr. Boonma received his B.E. in Computer Engineering from Chiangmai University, Thailand, and M.IT. in Computer Science from Monash University, Australia. He worked for the Department of Computer Engineering, Chaingmai University, as a faculty member since 1997. In 2001, Mr. Boonma received Australian Government Scholarship (AusAID) for his master program in Australia. In 2005, he received Thai Government Scholarship for his PhD. work in the United States.

Junichi Suzuki received the Ph.D. in computer science from Keio University, Japan, in 2001. Dr. Suzuki joined the Department of Computer Science, University of Massachusetts, Boston in September 2004, where he is currently an assistant professor. From 2001 to 2004, he was with the School of Information and Computer Science, University of California, Irvine (UCI), as a postdoctoral research fellow. Before joining UCI, he was with Object Management Group Japan, Inc., as the Technical Director. His research interests include autonomous adaptive distributed systems, biologically-inspired (e.g., ecological, genetic, immunological and developmental) software designs, and model-driven software engineering. In these areas, Dr. Suzuki has authored two books, published over 70 refereed papers including five award papers, gave over 100 invited talks, and served on technical program committees of over 25 conferences including IEEE AINA 2008 and IEEE/ACM/Create-Net/ICST BIONETICS 2007.

Dr. Suzuki is an active participant and contributor of the International Standard Organization SC7/WG19 and the Object Management Group, Super Distributed Objects SIG. He is a member of IEEE and ACM.

Please cite this article in press as: P. Boonma, J. Suzuki, BiSNET: A biologically-inspired middleware architecture ..., Comput. Netw. (2007), doi:10.1016/j.comnet.2007.06.006