Leveraging Biologically-inspired Mobile Agents Supporting Composite Needs of Reliability and Timeliness in Sensor Applications

Pruet Boonma$^{\#}$, Qi Han* and Junichi Suzuki$^{\#}$
$^{\#}$Department of Computer Science
University of Massachusetts, Boston
{pruet, jxs}@cs.umb.edu

*Department of Mathematical and Computer Sciences
Colorado School of Mines
qhan@mines.edu

Abstract

Wireless sensor networks are often used for event detection applications, which require a certain level of reliability and timeliness while minimizing energy consumption. Existing work considers reliability, timeliness and energy consumption largely in isolation. This paper proposes a solution to satisfy these conflicting requirements by using biologically-inspired mobile agents. The problem is formulated into an NP-hard problem, the Vehicle Routing Problem, and decentralized and centralized heuristics are developed to govern agent behaviors. Simulation results show that proposed solution allows agents (i.e., sensor applications) to effectively balance the tradeoffs among reliability, timeliness and energy efficiency and outperform an existing similar mechanism.

1. Introduction

Wireless sensor networks (WSNs) are often deployed to detect events that are distributed spatially such as fire spreading and oil spills. Due to the sheer number of sensor nodes and constant failures in the network, the detection of an event is often determined when a certain number of nodes report the same observation. For instance, a potential fire breakout may be identified when 80% of nodes report their temperature readings over 100 degrees. Without loss of generality, an event is identified when a certain percentage ($\alpha\%$) of nodes report their readings over a threshold. Furthermore, in order to enable a prompt response to the event, these sensor reports must reach the base station within a reasonable timeframe (D time units). Thus, a WSN application requires $\alpha\%$ of sensor reports within D time units for an event detection while minimizing energy consumption.

Existing work considers reliability, timeliness and energy consumption largely in isolation, as discussed in Section 5. Few attempts have been made to satisfy these requirements simultaneously. This simultaneous satisfaction imposes several challenges. First, reliability and timeliness are two competing goals. The requirement on reliability (i.e., the number of sensor reports) ensures that the base station can have enough information to make informed decisions on a detected event. The requirement on timeliness (i.e., deadline) aids timely decisions on a detected event. In order to ensure reliable data delivery, hop-by-hop recovery is often applied; however, this may not meet a given timeliness requirement. Second, reliability and energy efficiency conflict with each other. The more data the base station receives, the more reliable decisions can be made based on the data; however, more energy is consumed for extra data retransmissions and recovery actions. Third, there exists a tradeoff between timeliness and energy efficiency. In order to detect an event sooner, more energy is drained from nodes because more data transmissions are required.

This paper addresses the above challenge by designing WSN applications after biological systems. This design strategy is motivated by an observation that various biological systems have developed the mechanisms to meet conflicting requirements simultaneously. For example, a bee colony simultaneously maximizes the amount of collected nectar, maintains the temperature in a nest, and minimizes the number of dead drones [20]. If bees focus only on foraging, they fail to ventilate their nest and remove dead drones. Given this observation, this paper proposes a biologically-inspired architecture for WSN applications to adaptively balance the tradeoffs among conflicting requirements.

The proposed architecture models each WSN application as a group of multiple mobile agents. This is analogous to a bee colony (application) consisting of bees (agents). Agents read/collect sensor data (as nectar) on individual nodes (modeled as flowers), and carry (or push) the data through multiple hops to the base station, which is modeled as a nest of bees. If they do not satisfy a desired level of reliability (i.e., the number of sensor data required for an event detection), extra agents leave the base station (nest) to the network for collecting (or pulling) extra sensor data from nodes. Agents perform these push/pull functionalities by invoking biologically-inspired behaviors such as migration, swarm formation and replication.

In order for agents to optimally perform their behaviors in terms of reliability, timeliness and energy efficiency, agent behaviors are formulated into a well-known NP-hard problem, the Vehicle Routing Problem (VRP). Agents perform a decentralized and centralized VRP heuristics to push and pull sensor data, respectively. Simulation results show that the VRP-formulated migration behavior allows agents (i.e., WSN applications) to adaptively balance the tradeoffs among reliability, timeliness and energy efficiency and outperform an existing similar mechanism.

This paper is organized as follows. Section 2 states and formulates a problem this paper addresses. Section 3 proposes VRP heuristics for the agent migration behavior. Section 4 shows a series of simulation results to evaluate the proposed VRP-formulated migration behavior. Sections 5 and 6 conclude with some discussion on related work.

2. Problem Statement and Formulation

This section states and formulates the problem that this paper addresses.

2.1. Problem Statement

This paper assumes WSN applications, each of which requires the base station to collect at least N_R sensor data within D time units. N_R is referred as the desired reliability. N_{rd} (the actual reliability) denotes the actual number of data received by the deadline. In order to reliably detect an event, $N_{rd} \geq N_R$. In other words, each WSN application requires the normalized reliability $\frac{N_{rd}}{N_R} \geq 1$ while minimizing energy consumption.

In order to formally state the problem at hand, this paper uses the following notations to describe WSNs. A WSN is considered as a graph $G(V, E)$.

- $V = \{v_0, v_1, ..., v_n\}$ is a vertex set, where v_0 is the base station. $V' = V - \{v_0\}$ is a set of n sensor nodes. Each node periodically generates sensor data.
- $E = \{(v_i, v_j) | v_i, v_j \in V; i \neq j\}$ is an edge set. An edge is established from the node v_i to v_j if v_i can transmit a packet to v_j. Due to the nature of asymmetric communication in WSNs, an edge is directed; $(v_i, v_j) \in E$ does not necessarily mean $(v_j, v_i) \in E$.
- c_{ij} is a non-negative weight associated with the edge (v_i, v_j). It represents the cost for moving an agent between the nodes v_i and v_j. Section 2.3 describes the cost function to determine c_{ij}.
- t_{ij} is the latency for an agent to move from the node v_i to v_j.
- m is the number of agents. Each agent can carry a limited size S of data due to the limitation of packet size. This is a constraint on how many nodes an agent can collect data from.
- R_k is a migration route for the agent k to follow. C_{R_k} is the cost of moving the agent k along the route R_k. $C_{R_K} = \sum_{(h,h') \in R_k} c_{hh'}$; h' is the next hop node of the node h in the route R_k.
- T_{R_k} is the latency for the agent k to move along the route R_k. $T_{R_k} = \sum_{(h,h') \in R_k} t_{hh'}$.

The problem this paper addresses is to, given a set of n nodes, determine a set of m agents that can satisfy $\frac{N_{rd}}{N_R} \geq 1$ and the migration route (R_k) of each agent such that $\sum C_{R_k}$ is minimized subject to $\max T_{R_k} \leq D$.

2.2. Problem Formulation with VRP

VRP can be described as follows. Let there be n demand points in a given area, each demanding a quantity of weight Q_i ($i = 1, 2, \cdots, n$) of goods to be delivered to it. The goods are stored at a depot, where a fleet of vehicles is stationed. Vehicles have the identical maximum weight capacity and maximum route time (or distance) constraints. They must all start and finish their routes at the depot. It is assumed that Q_i is less than the maximum weight capacity of each vehicle and Q_i is delivered by a single vehicle.

In VRP, both the required number of vehicles and their routes are unknown. The objective of VRP is to obtain a set of routes for vehicles to minimize their total route time. In fact, VRP is an m-TSP problem with two additional constraints: the maximum weight capacity and maximum route time for each vehicle.

This paper reduces the problem stated in Section 2.1 to VRP. In the problem at hand, there are n sensor nodes (demand points) in the network. Each node v_i has a sensor data of size l_i bytes to be delivered to the base station (the depot) by an agent (an vehicle). The packet size limitation in WSNs is analogous to the vehicle weight capacity in VRP. The timeliness constraint in WSNs is mapped to the maximum vehicle route time in VRP.

2.3. Cost Function

This section defines the function to determine the link cost between the node v_i and v_j (c_{ij}; See Section 2.1.). This paper uses packet loss rate to determine link cost.

To avoid the asymmetric nature of communication links, the link cost c_{ij} is determined as $f_{ij} \times f_{ji}$, where f_{ij} is the loss rate to transmit packets (agents) from the node v_i to v_j. Packet loss rate simultaneously impacts the reliability, timeliness and energy efficiency of sensor data transmission (agent transmission). Lower packet loss rate better meets all of the three requirements.

Packet loss rate is measured when nodes are deployed[1]. Each node transmits a set of packets to each neighboring

---

[1] Currently, assuming that WSNs are semi-static [24, 17, 28], packet

node. Each packet contains its sequence number and the total number of transmitted packets. Upon receiving a set of packets, each neighboring node determines packet loss rate based on the number of received packets.

3. Biologically-inspired Mobile Agents

In order to solve the problem at hand, this paper proposes to use biologically-inspired mobile agents in a push and pull hybrid manner. There are two types of agents: *event agents* and *query agents*. An event agent (EA) is deployed on each node. It carries (or *pushes*) a sensor data to the base station using multiple hops. On its way to the base station, each EA swarms with other EAs to aggregate as many sensor data as possible as long as it meets a given deadline. Due to inherent failures in WSNs, EAs may not be able to satisfy the desired reliability (the number of sensor data required for an event detection). In this case, query agents (QAs) are created at the base station and dispatched to the network for collecting (or *pulling*) missing sensor data from nodes.

3.1. Biologically-inspired Agent Behaviors

Agents (EAs and QAs) implement the following behaviors.

Replication: Agents (EAs and QAs) may make a copy of themselves. An EA replicates itself on a node when it detects an event of interest, which is application-specific and may simply be a sensor reading exceeding a threshold. A replicated EA contains collected sensor data can carries to the base station. A QA is replicated at the base station and dispatched to the network to collect sensor data from nodes.

Swarming: Agents (EAs and QAs) may swarm (or merge) with other agents on their way to the base station. EAs swarm with other EAs, and QAs swarm with other QAs. With this behavior, multiple agents become a single agent. The resulting (swarm) agent aggregates sensor data. This data aggregation saves power consumption of nodes because in-node data processing requires much less power consumption than data transmission does.

Migration Agents may move from one node to another. Migration is used to deliver agents (sensor data) to the base station. There are two ways for agents to move.

- *Chemotaxis walk:* The base station periodically propagates *base station pheromones* to individual nodes in the network. Their concentration decays on a hop-by-hop basis. (Each pheromone evaporates in a certain time period.) Agents (EAs and QAs) can locate the base station approximately, and move to the base station in the shortest paths by sensing pheromone's concentration gradient[2].
- *Sidestep walk:* In addition to the chemotaxis walk, each EA may sidestep the shortest migration path and move to a neighboring node that has the equal or longer distance to the base station, as long as the EA meets a given deadline to reach the base station. This behavior encourages EAs to perform swarming-based data aggregation by increasing the number of nodes EAs visit. QAs are not allowed to perform this behavior.

Agents perform their behaviors with VRP heuristics. This paper proposes a decentralized VRP heuristics for EAs, and leverages an existing centralized VRP heuristics for QAs. Particularly, these VRP heuristics are used to answer the following questions.

- Where and how should EAs replicate themselves?
- How many agents (EAs and QAs) should be created?
- How should each agent (EA and QA) move?

3.2. A Decentralized VRP Heuristics for Event Agents

EAs implement a decentralized VRP heuristics to carry sensor data to the base station by a given deadline. To the best of the authors' knowledge, there is no existing heuristics to solve VRP in a decentralized way. This paper proposes a decentralized greedy algorithm to govern the EA behaviors.

The proposed algorithm uses a cluster-based approach to determine where and how EAs replicate themselves. Nodes are grouped to form clusters, and an EA replicates itself on each cluster head when it detects an event. Each cluster has one-hop topological radius, and all neighboring nodes of a cluster head become its cluster members.

Cluster head election is designed to maximize the number of cluster members by choosing a sensor node who has many neighboring nodes. In this process, each node becomes idle first for T_{idle} time units. It calculates T_{idle} by randomly choosing a number between zero and T_{max}/N. T_{max} is a constant that specifies the bound of cluster head election period, and N is the number of neighboring nodes. After this idle period, each node becomes a cluster head and broadcasts an ADV (advertisement) message to its neighboring nodes. However, if a node receives an ADV message from any of its neighboring nodes during the idle period, it becomes a cluster member of the node who originates the ADV message. Each cluster member sends a JOIN message to its cluster head so that the cluster head know who are cluster members. Through this process, clusters are uniformly distributed and cover the entire network. Note that each node always belongs to a single cluster; if it receives multiple ADV messages during its idle period, it responds to the first ADV message and ignores subsequent ones.

---

loss rate is measured at the beginning of a WSN operation. It can be periodically measured and updated; however, it is out of this paper's scope.

[2] Base station pheromones are designed after the Nasonov gland pheromone, which guides bees to move toward their nest [7].

When an EA detects an event on a cluster head, the EA replicates itself one or more times. The replicated EAs visit cluster members to collect sensor data from them. This way, each EA aggregates sensor data and carries the aggregated data to the base station. The ideal number of replicated EAs per cluster is $\lceil \frac{n}{s} \rceil$, where n is the expected number of nodes in a cluster and s is the number of data that a single EA can carry. If an EA already contains s number of data and cannot contain any more, the EA is refereed as a *fat* EA. If an EA can still contain data, it is referred as a *slim* EA.

Each fat EA moves toward the base station on a hop by hop basis by selecting the next hop node that minimizes the link cost (c_{ij} in Section 2.3). This allows fat EAs to increase the chances to reach the base station by a given deadline.

By default, each slim EA also chooses the next hop node that minimizes link cost as well. However, when it finds a cluster on its way to the base station and has not visited the cluster's head node, the EA sidesteps to the cluster head for swarming with other slim EAs as as far as it meets a timeliness constraint. If there is no slim EAs on the cluster head, the EA stays there for a period of time before moving to the base station again. This period increases the chances for a waiting EA to swarm with other slim EAs while allowing it to reach the base station within a given time constraint.

The waiting period of each slim EA is calculated by each cluster head based on a given deadline and the latency from the cluster head to the base station. Let T_d be the deadline, and $t_{i,b}$ be the latency from the cluster head i to the base station, a slim EA at cluster head i can wait for $T_d - t_{i,b}$ before it starts moving towards the base station. This waiting time allows slim EA to move to the base station within the deadline, as long as the deadline is greater than the longest traveling time. In addition, the waiting time allows slim EAs to increase the chance to combine with other slim EAs. For instance, we assume that on its way to the base station, a slim EA at cluster head i has to visit cluster head j which also has a slim EA. Let $t_{i,b}$ and $t_{j,b}$ be the latency from the cluster head i and j to the base station respectively. The traveling time from the cluster head i to j, $t_{i,j}$, is then approximately $t_{i,b} - t_{j,b}$. The slim EA at cluster head i will wait until $T_d - t_{i,b}$, while slim EA at cluster head j will wait until $T_d - t_{j,b}$. When slim EA at cluster head i starts moving at $T_d - t_{i,b}$, it will reach cluster head j at time $T_d - t_{i,b} + t_{i,j}$. This is the same as the time that slim EA in cluster head j is supposed to leave, which is $T_d - t_{j,b}$. So, the two EAs will combine and then leave cluster head j. This waiting and combination process is performed repeatedly along the way to the base station. In practice, the waiting time can be considered as an upper bound instead of a hard deadline. Therefore,an EA may leave a cluster head before the waiting time expires.

3.3. A Centralized VRP Heuristics for Query Agents

QAs implement a centralized VRP heuristics to visit a certain number of nodes from the base station and collect extra sensor data on the nodes. To find an optimal number of QAs and also traveling path of each QA, Clarke-Wright Savings algorithm [3, 12], a well known VRP solving algorithm, is used with some modifications. The Clarke-Wright Savings algorithm is an heuristic algorithm which uses constructive methods to gradually create a feasible solution with modest computing cost. Basically, the Clarke-Wright Savings algorithm starts by assigning one agent per vertex in the graph. The algorithm then tries to combine two routes so that an agent will serve two vertices. The algorithm calculates the "savings" of every pair of routes, where the savings is the reduced total link cost of an agent after a pair of route is combined. The pair of routes that have the highest saving will then be combined if no constraint, time or capacity, is violated.

In this paper, Clarke-Wright Savings algorithm is extended to consider the time and space constraint. By looking into the data the base station has received from the EAs, the base station can determine to which cluster or area a QA should be dispatched initially.

- An *internal path*, R_j, is created within each cluster, X_j which sensor readings are missing. Consider a set of node $\{v | v \in X_j\}$, Clarke-Wright Saving can be used by choosing a cluster head, i.e. swarm location, $\hat{v}_j$ as a depot, then create a path to visit every $v \in X_j - \{\hat{v}_j\}$. The time, t_j, to travel within the cluster is also assigned to the cluster.
- The cluster head, $\hat{v}_i$, is selected from the cluster X_j to represents the location of the cluster.
- The shortest route $R_i j$ between two nodes, $\hat{v}_i$ and $\hat{v}_j$ where $i \neq j$ are calculated using Floyd-Warshall algorithm. The distance between nodes are measured by cost, $\hat{c}_{ij}$ of moving agent between two nodes, which is the function of packet loss rate.
- A route R_{0j} is created from base station to each node $\hat{v}_j$.
- The saving of combining a pair of routes between the base station and two individual nodes($\hat{v}_j$; cluster representative) are computed.

  $$s_{ij} = \hat{c}_{0i} + \hat{c}_{0j} - \hat{c}_{ij} \tag{1}$$

  The saving must obey two constraints; first, the traveling time along the combining route must less than deadline, $t_{0i} + t_i + t_{ij} + t_j + t_{j0} < D$ and the number of node in the route R_{0ij}, $|X_i| + |X_j|$, is less than space limit, S.
- The saving is ordered from the largest to smallest into a saving list
- Begin at the top of the saving list, a sub-tour is formed by merging the routes, R_{0i} and R_{0j}, that create the saving, s_{ij};

  - a new route, R_{0ij} is constructed with traveling cost $\hat{c}_{0ij}$ and time t_{0ij}.
  - the route R_{0i} and R_{0j} are removed.
- The process is repeated from the first step until no more possible saving.

Finally, a set of routes between cluster are constructed and an QA is assigned for each route. Also, the traveling route inside each cluster is given to a QA who is going to visit the cluster. Then, QAs are dispatched to collect data from each cluster by visiting the cluster head first.If QA can visit cluster head and the cluster head still have the sensor readings from each cluster members, QA can collect sensor readings from the cluster head and travel back to the base station immediately. However, if QA cannot visit the cluster head, e.g. cluster head is missing or running out of battery, QA then consult the traveling path inside the cluster which assigned by base station to visiting each cluster member to collect data and then travel back to base station.

4. Performance Evaluation

The proposed approach is implemented in NesC and evaluated using TOSSIM 1.0 [13]. A sensor network is simulated in an area of 200x200 square meters. In most of our experiments, the network consists of 150 sensor nodes modeled after MICAz mote with communication radius of about 30 meters, bandwidth of approximately 200kbps and 128kB of memory space[4]. B-MAC is used as the MAC layer protocol by using CC2420 radio module in TinyOS. The sensor nodes are uniformly deployed in the area.

To the best of our knowledge, only MMSPEED satisfies reliability and timeliness requirements simultaneously [5]. MMSPEED provides active on-time reachability of packets by using multiple speed levels and multi-path routing. It uses SPEED [10] for the timeliness guarantee and adds probabilistic reliability guarantee based on probabilities of reliable delivery of packets at different links. MMSPEED provides the flexibility for applications to choose several different levels of reliability and timeliness. However, it does not consider minimizing the energy consumption in routing. Therefore, we implemented MMSPEED in TinyOS for comparison. In addition, we use TinyOS's Drain Data Collection Protocol [22] as a baseline. The application's desired reliability is varied from 0.6 to 1.0 and desired freshness, which is a metric to measure timeliness, is varied from 60 and 100 seconds. Each sensor node reports sensing values one at a time and each agent can carry up to 10 readings. We will evaluate the system performance to demonstrate how proposed approach achieves the desired reliability, freshness and the energy consumption involved. We will also study how the network density, the packet size, i.e. the maximum number of sensor readings an agent can carry, and network fault severity affect the system performance.

Figure 1 shows the actual reliability against the desired reliability when desired reliability is set to be 0.5 and 1.0. The results show that for proposed approach, a reliability of 0.74 can be achieved by purely using EAs. When the desired reliability is greater than 0.74 (i.e., 1.0), QAs are dispatched to collect additional data in order to archive higher reliability. However, due to the time constraint imposed by the application, the actual reliability can be less than the desired reliability. For example, when the freshness is very short (60 seconds in the figure), the highest achievable actual reliability is 0.81, which may be lower than the desired reliability. However, if the freshness is long enough, i.e. 90 - 100 seconds, the actual reliability can be equal or higher than the desired reliability. In contrast, Drain and MMSPEED can not improve the actual reliability beyond 0.73 and 0.85 respectively because both of them rely purely on push mechanism. Figure 2 shows the total energy consumption using the same parameter settings as in Figure 1. When the desired reliability is greater than 0.7, data collected by EAs can not satisfy the desired reliability, so the QAs are dispatched to gather additional data; hence, the increase in the total energy consumption. Compared with Drain and MMSPEED which consume similar amount of energy irrespective of any reliability requirement, proposed approach consume less energy when the desired reliability is low, i.e. 0.5. Moreover, when the desired reliability is high, i.e. 1.0, proposed approach has lower energy consumption due to the data aggregation mechanism used in proposed approach.

Figure 3 shows the impact of desired freshness on the actual freshness when desired freshness is set to be 60 seconds and 100 seconds. As shown in the figure, when desired reliability is low (less than 0.8), the actual freshness is always less than the desired freshness. This is because the EAs can collect data from the network without the help from QAs, which use additional time to collect data. However, when the desired reliability is increased to 0.8 or 0.9, the actual freshness may be higher than the desired freshness. In the extreme case, i.e., the desired reliability is 1.0, EAs and QAs are both used to collect data to satisfy the desired reliability; therefore, the actual freshness is very high. Comparing to proposed approach, the actual freshness of Drain is always flat because it relies purely on push mechanism, which results in low actual reliability. MMSPEED, on the other hand, tries to match actual freshness with desired freshness. However, from figure 4, it is clearly that the energy consumption of MMSPEED is constant regarding the desired reliability. Nevertheless, proposed approach can reduce the energy consumption when the desired reliability is low, i.e. by using only EAs.

Figure 5 depicts the relationship between the node density in a network and the ratio of actual reliability over desired reliability. The desired reliability is 0.9 here and the

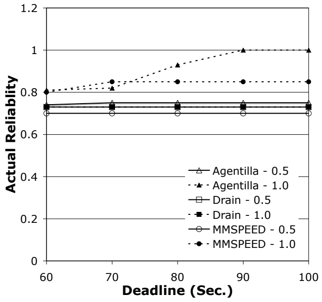

Figure 1. Impact of desired reliability on the actual reliability with varied deadline

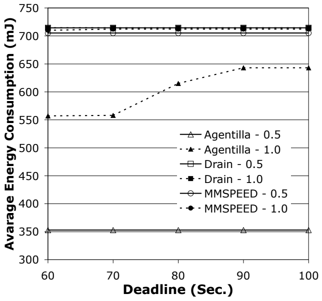

Figure 2. Impact of desired reliability on energy consumption with varied deadline

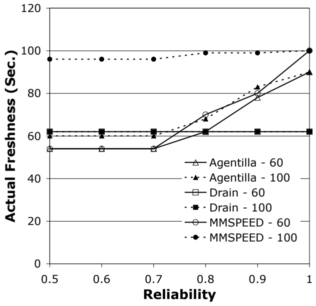

Figure 3. Impact of desired reliability on the actual freshness with varied desired reliability

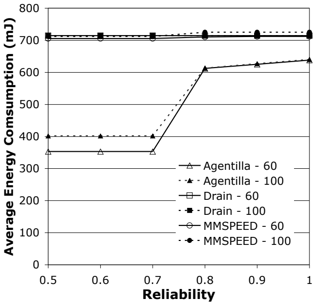

Figure 4. Impact of desired freshness on energy consumption with desired reliability

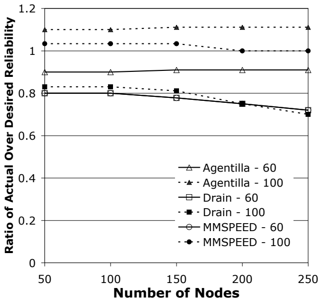

Figure 5. Ratio of actual reliability over desired reliability with different node density

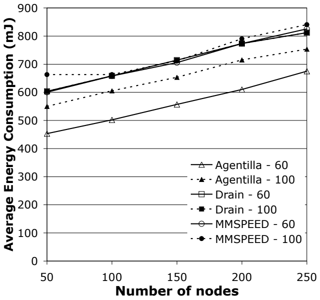

Figure 6. Average energy consumption with different node density

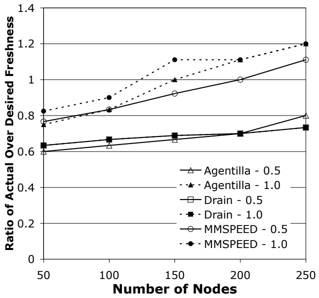

Figure 7. Ratio of actual reliability over desired freshness with different node density

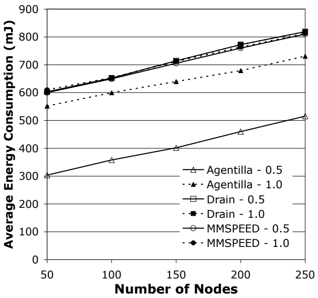

Figure 8. Average energy consumption with different node density

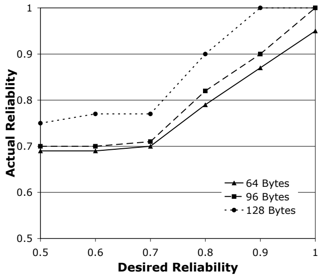

Figure 9. Impact of desired reliability on the actual reliability with varied maximum packet size

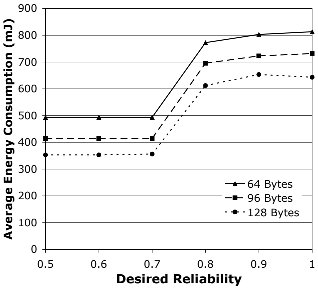

Figure 10. Impact of desired reliability on energy consumption with varied maximum packet size

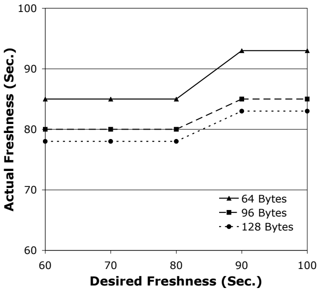

Figure 11. Impact of desired freshness on the actual freshness with varied maximum packet size

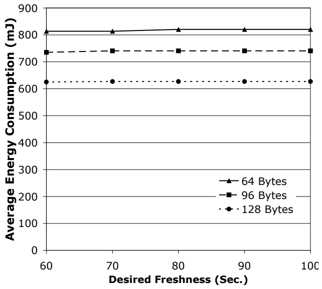

Figure 12. Impact of desired freshness on energy consumption with varied maximum packet size

desired freshness is set to be 60 and 100 seconds. It is observed that node density has very small effect on proposed approach due to the decentralized nature of the protocols used in proposed approach. On the other hand, the reliability ratio of Drain and MMSPEED are slightly decreased because of increased overhead in topology management when the network size is increased. Figure 6 shows the average energy consumption per sensor node in different node density under the same settings as used in Figure 5. When the network size varies from 50 to 250 nodes, the energy consumption is linearly increased because the amount of data that have to be carried to the base station is proportional to the number of node. Therefore, in terms of energy consumption, the proposed algorithm scales well when the number of sensor nodes increases. This conclusion is also applied to Drain and MMSPEED.

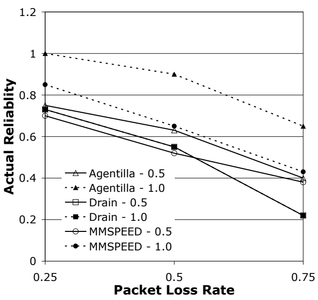

Figure 13. Impact of desired reliability on the actual reliability with varied packet loss rate

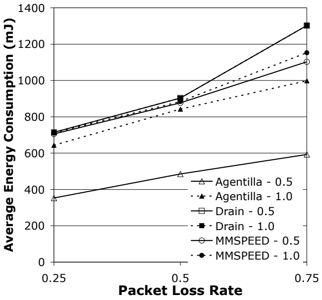

Figure 14. Impact of desired reliability on energy consumption with varied packet loss rate

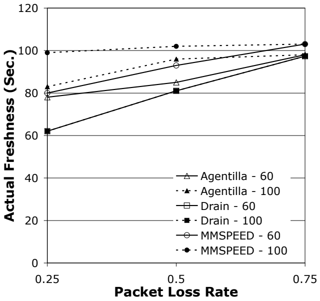

Figure 15. Impact of desired freshness on the actual freshness with varied packet loss rate

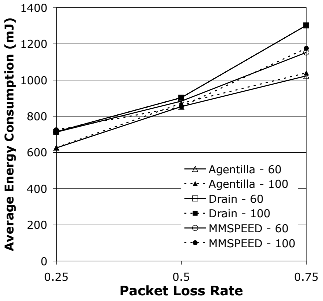

Figure 16. Impact of desired freshness on energy consumption with varied packet loss rate

Figure 7 represents the relationship between the node density in a network and the ratio of actual freshness over desired freshness where the desired freshness is set to be 90 seconds. The desired reliability is set to 0.5 and 1.0. The ratio of actual over desired freshness are slightly increased when the number of sensor nodes is increased, mainly because of increasing traffic among sensor nodes. Figure 8 is similar to the result in Figure 6. Therefore, the proposed architecture scales well when the node density increases. Moreover, the energy consumption of proposed approach is always lower than Drain and MMSPEED, especially when the desired reliability is low, i.e. 0.5, thanks to the hybrid approach, i.e. QAs is used only when the data from EAs cannot meet the desired reliability, and data aggregation mechanism in proposed approach.

Figure 9 shows the actual reliability against the desired reliability. The maximum packet size is set to be 64, 96 and 128 bytes, which allows an agent to carried at most 5, 8 and 10 sensor reading respectively. The results indicate that the packet size has some impact on the actual reliability; however, proposed approach is still able to meet the desired reliability. Figure 10 shows the similar trend as Figure 9. Figure 11 and Figure 12 demonstrate the impact of desired freshness as the maximum packet size varies. Similar trends are observed as shown in Figure 9 and Figure 10.

Figure 13 shows the impact of packet loss rate on the actual reliability. The desired reliability is set to be 0.5 and 1.0. As shown in the figure, when loss rate is increased, the actual reliability is decreased; however, proposed approach can archive higher actual reliability than Drain and MMSPEED because proposed approach can reduce the number of packets sending to the network using data aggregation, which leads to less chance to lose a packet. Figure 14 shows the similar results as Figure 13. Figure 15 and Figure 16 depicts the impact of desired freshness when packet loss rate varies. The trend observed from these figures is similar to that in Figure 13 and Figure 14.

To summarize, our experimental results show that the hybrid push (achieved by EAs) and pull (performed by QAs) approach can meet user-specified reliability and freshness with reasonable amount of energy consumption. proposed approach outperforms Drain or MMSPEED, especially in terms of energy efficiency.

5. Related Work

Reliability and timeliness have been mostly investigated separately in WSNs. For instance, Gradient Broadcast proposes a multi-path routing method to improve the reliability of data transmission [27]. PSFQ addresses reliable packet transmission via hop-by-hop recovery [23]. ESRT and PERG control transmission frequency and retransmission rate to yield the optimal reliability of data transmission [19, 9]. RAP addresses the timeliness requirement by prioritizing packets and supporting priority queues in the MAC layer [15]. SPEED also proposes a custom timeliness-aware MAC protocol. Unlike these work, the proposed mechanism focuses on satisfying the reliability and timeliness requirements simultaneously.

Mobile agents have been used in WSNs for routing [8, 25], dynamic application re-programmability [6, 21], security [14], network exploration [16] and data aggregation/dissemination [26, 2, 11]. However, none of them address the simultaneous satisfaction of reliability and timeliness. MADSN allows agents to collect and aggregate sensor data from particular regions based on the itineraries assigned by the base station [18]. However, each itinerary is not optimized against energy consumption. In contrast, the proposed mechanism aims to minimize energy consumption while meeting the reliability and timeliness requirements.

Similar to the proposed mechanism, [17] proposes several algorithms to produce a routing path for a packet to visit a certain set of nodes and collect data from the nodes.

A produced path is optimized against energy consumption within a constant factor of the optimum. [17] considers the issue of fixed-size packets as well as node/link failures; however, it does not address the timeliness requirement.

6. Conclusion

This paper considers WSN applications required to simultaneously satisfy a certain level of reliability and timeliness while minimizing energy consumption. The proposed algorithms are designed to balance these conflicting requirements by using biologically-inspired mobile agents in a push-pull hybrid manner. EAs migrate one node to another and carry (or push) sensor data to the base station. When EAs do not satisfy a desired level of reliability (a desired number of data reported to the base station), the base station dispatches QAs to the network for collecting (or pulling) extra data from nodes. The behaviors of EAs and QAs are formulated into the Vehicle Routing Problem (VRP). In order to optimally behave in term of timeliness and energy efficiency, EAs and QAs perform a decentralized and centralized VRP heuristics, respectively. Simulation results validate that the VRP-formulated migration behavior allows agents (WSN applications) to adaptively balance the tradeoffs among reliability, timeliness and energy efficiency and outperform an existing similar mechanism called MMSPEED.

References

[1] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5), 2005.

[2] M. Chen, T. Kwon, and Y. Choi. Data dissemination based on mobile agent in wireless sensor networks. In *Proc. of the IEEE Conf. on Local Computer Networks*, 2005.

[3] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *INFORMS Operations Research*, 12(4):568–581, 1964.

[4] Crossbow Technology Inc. Micaz datasheet, 2006.

[5] E. Felemban, C. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic qos guarantee in reliability and timeliness domains in wireless sensor networks. In *Proc. of the 24th Annual IEEE Conf. on Computer Communications*, 2005.

[6] C.-L. Fok, G.-C. Roman, and C. Lu. Mobile agent middleware for sensor networks: An application case study. In *Proc. of Int'l Conf. on Info. Processing in Sensor Nets.*, 2005.

[7] J. B. Free and I. H. Williams. The role of the nasonov gland pheromone in crop communication by honey bees. *Int'l J. of Behavioural Biology*, 41:314–318, 1972.

[8] L. Gan, J. Liu, and X. Jin. Agent-based, energy efficient routing in sensor networks. In *Proc. of the 3rd Int'l Joint Conf. on Autonomous Agents and Multiagent Systems*, 2004.

[9] Q. Han, I. Lazaridis, S. Mehrotra, and N. Venkatasubramanian. Sensor data collection with expected reliability guarantees. In *Proc. of the 1st IEEE Int'l Workshop on Sensor Networks and Systems for Pervasive Computing*, 2005.

[10] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proc. of the 23rd IEEE Int'l Conf. on Distributed Computing Systems*, 2003.

[11] S. Hussain, E. Shakshuki, A. W. Matin, and A. R. Matin. Collaborative agents for data dissemination in wireless sensor networks. In *Proc. of the 20th Int'l Sympo. on High-Perf. Comp. in an Advanced Collaborative Env.*, 2006.

[12] J. K. Lenstra and A. R. Kan. Complexity of vehicle routing and scheduling problems. *Wiley Networks*, 11(2), 1981.

[13] P. Levis. Tossim: Accurate and scalable simulation of entire tinyos applications. In *Proc. of the 1st ACM Conf. on Embedded Networked Sensor Systems*, 2003.

[14] Z. Liu and Y. Wang. A secure agent architecture for sensor networks. In *Proc. of Int'l Conf. on AI*, 2003.

[15] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *Proc. of IEEE Real-Time and Embedded Tech. and App. Symposium*, 2002.

[16] D. Massaguer, C.-L. Fok, N. Venkatasubramanian, G.-C. Roman, and C. Lu. Exploring sensor networks using mobile agents. In *Proc. of the 5th Int'l Joint Conf. on Autonomous agents and multiagent systems*, 2006.

[17] A. Meliou, D. Chu, C. Guestrina, J. Hellerstein, and W. Hong. Data gathering tours in sensor networks. In *Proc. of Int'l Conf. on Info Processing in Sensor Nets.*, 2006.

[18] H. Qi, S. Iyengar, and K. Chakrabarty. Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 31(3):383–391, 2001.

[19] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz. Esrt: Event-to-sink reliable transport in wireless sensor networks. In *Proc. of the 4th ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing*, 2003.

[20] T. Seeley. *The Wisdom of the Hive*. Harvard Uni. Press, 2005.

[21] L. Szumel, J. LeBrun, and J. D. Owens. Towards a mobile agent framework for sensor networks. In *Proc. of the 2nd IEEE Workshop on Embedded Networked Sensors*, 2005.

[22] G. Tolle and D. Culler. Design of an application-cooperative management system for wireless sensor networks. In *Proc. of European Workshop on Wireless Sensor Work.*, 2005.

[23] C. Wan, A. Campbell, and L. Krishnamurthy. Pump slowly fetch quickly(psfq): A reliable transport protocol for wireless sensor networks. *IEEE J. Sel. Areas Comm.*, 23(4), 2005.

[24] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. Int'l Conf. on Embedded Networked Sensor Sys.*, 2003.

[25] Y. Xu and H. Qi. Mobile agent migration algorithms for collaborative processing. In *Proc. of the IEEE Wireless Communications and Networking Conf.*, 2006.

[26] Y. Xu, H. Qi, and P. T. Kuruganti. Mobile-agent-based computing model for collaborative processing in sensor networks. In *Proc. of the IEEE Global Telecom. Conf.*, 2003.

[27] F. Ye, G. Zhong, S. Lu, and L. Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks*, 11(3), 2005.

[28] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proc. of Int'l Conf. on Embedded Networked Sensor Sys.*, 2003.