Abstract

Wireless Sensor Networks (WSNs) are severely constrained in computation and communication capabilities due to the cost and size of available sensors. On the other hand, autonomic computing (AC) offers a promising solution to manage large-scale computing systems without human intervention. Realizing the similarity between WSNs and AC applications, this paper proposes an autonomic sensor network framework to enable self-managing wireless sensor network systems for collaborative information processing. In particular, a preliminary power-aware self-configuring and self-optimizing sensor selection scheme is developed to improve the performance and extend the lifetime of sensor networks. The simulation results confirm that the proposed power-aware scheme prolongs the network lifetime and balances the energy in sensor nodes.

1. Introduction

The emerging pervasive wireless sensor networks (WSNs) have ushered in a wide spectrum of novel applications in science, engineering, business, and military. Sensors with the capabilities of sensing, computation, processing, and communication are deployed in certain environments widely and unobtrusively to accomplish high-level tasks such as object tracking, smart space, environmental monitoring, and homeland security [5] [6] [13] [28]. Due to the size and cost of sensor nodes, it is unfeasible to equip sensor nodes at a large scale with high energy and powerful computational capability in real-world applications, compared with hand-held devices such as PDAs and cell phones. On the other hand, networking large number of sensor nodes (thousands or millions of sensors) requires the whole system to accomplish sensing efficiently and accurately. As a result, how to use these vast collective resources and extend the lifetime of a sensor network become the most challenging issue in designing sensor networks. The current trend is to take advantage of in-network collaborative information processing to reduce the expensive communication cost and minimize the processing time [29]. However, the dynamism – sensor nodes wake up, sleep, and die dynamically, the heterogeneity – different sensor nodes can be deployed in an incremental fashion to maintain the long-term functionalities of WSNs, and the limited power, computing, and communication capabilities, pose significant challenges for managing WSNs in an efficient and holistic manner.

An emerging paradigm, autonomic computing, offers a promising solution to manage sensor networks by enabling system-wide holistic self-management. Inspired by biological systems, the autonomic computing paradigm features self-configuration, self-healing, self-optimization, and self-protection, coined as self-CHOP [11] [14] [20]. To enable self-manageability, autonomic computing follows a goal-oriented approach. Therefore an autonomic computing starts with the input of high-level requirements. It then goes through the requirement analysis process that analyzes and models the requirement to decompose and formulate policies. Moreover, autonomic computing aims to support adaptive functionalities by its four key features [8] [14] [20]. It is constructed from a large number of autonomic elements with self-CHOP functionalities. An autonomic element consists of sensors, actuators, and specialized knowledge-base for particular managed elements. They collaboratively regulate behaviors using the built-in self-CHOP functionalities and knowledge base.
Autonomic computing technologies provide a potential means to address challenging issues in managing WSNs [27]. There are two important similarities between autonomic systems and WSNs, i.e., networked autonomic elements vs sensor nodes and high-level goals-oriented. In particular, autonomic computing can be employed to improve collaborative information processing and data aggregation in WSNs so that expensive communication costs of large amount of data can be substantially reduced and the system-wide self-management is achieved.

In this paper, we propose a novel managing framework, Autonomic Sensor Networks, integrating four key characteristics into wireless sensor networks to improve the collaborative information processing task in a systematic manner. To support such a self-manageable network, we develop a power-aware sensor selection scheme with the self-optimizing capability. Sensors to participate in certain task are chosen adaptively to improve the sensing quality and prolong the network lifetime.

The rest of the paper is organized as follows. In Section 2, the current implementation patterns of autonomic computing systems are classified into different categories and we introduce the previous work in cluster organization and sensor selection in wireless sensor networks. A conceptual architecture of autonomic sensor network and a power-aware routing scheme are presented in Section 3. Following that, Section 4 presents the experimental evaluation through simulation. We conclude in Section 5 with the avenue of future work.

2. Related Work

2.1. Autonomic Computing Paradigm

Before the proposed architecture of autonomic sensor networks is presented, it is important to classify the existing autonomic paradigms. Following autonomic systems have been proposed such as component-based, recovery-oriented computing, aspect-oriented programming, multi-agent systems, peer-to-peer, and service-oriented computing paradigms. These implementation patterns may have overlapped features.

Component-based paradigm – A natural paradigm to realize the autonomic system architecture is to adopt a component-based implementation. Each component is equipped with self-contained policies and mechanism [17] [12] [21]. The Accord component-based programming model in the AutoMate project [17] [19] realizes three fundamental separations: 1) a separation of computations from coordination and interactions; 2) a separation of nonfunctional aspects (e.g., resource requirements, performance) from functional aspects; and 3) a separation of policy and mechanism. An autonomic element in AutoMate exposes three sets of interfaces: functional, control, and operational ports. Smart components [12] can adapt to environmental changes to sustain high performance.

Service-oriented paradigm – Service-oriented architecture gains strong support from academia and industry owing to its open standard and interoperability [10] [30]. This category covers grid, peer-to-peer, and web service technologies as they are converging in many aspects.

Recovery-oriented computing paradigm – Recovery-oriented computing (ROC) [31] aims to build highly-dependable Internet services by emphasizing recovery from failure rather than failure-tolerance or failure-avoidance. It largely falls into the category of self-healing characteristic of autonomic systems. Primary research areas in ROC include isolation and redundancy, system-wide support for undo, integrated diagnostic support, online verification of recovery mechanisms, design for high modularity, measurability, restartability, and dependability/availability benchmarking. For example, proactive restarting components or software rejuvenating can improve overall system availability and software aging problems.

Multi-agent paradigm – A rational agent is anything that can perceive its environment through sensors and act upon that environment through actuators [22]. Every single agent has limited functionalities. To collaboratively work towards some objectives, agents are grouped together to form multi-agent systems (MAS). The MAS model has been used to realize self-organizing systems and support distributed collaborative working environments [24]. It bears some common characteristics with the component-based paradigm.

2.2. Organization and sensor selection

Basically there are two kinds of power consumption in wireless sensor networks, i.e., computation and communication. Particularly, the ratio of energy consumption for communication and computation is typically in the scale of 1000. The redundant information provided by densely deployed sensor nodes requires the number of sensors to participate in the task much more than necessity, which severely affects the network lifetime [4].
Several methods have been proposed recently to reduce power consumption caused by communication [2] [7] [16] [23] [26]. Local greedy algorithms have been developed by [26] and [2] to select the next most informative sensor node based on entropy and information utility respectively. The problem, however, is that if the optimal path is always chosen, the nodes along the path will deplete energy more quickly than others, which leads to the network partition. Instead, Shah and Rabaey [23] they proposed that sometimes sub-optimal paths should be chosen depending on the probabilities to elongate the whole network lifetime. On the other hand, the accuracy of target status can not be guaranteed and it adds latency and computation load by tracing back the path to store the average cost by their algorithm. Gupta et al. dealt with the sensor scheduling stochastically in [7] as well. Another algorithm called “max-min $zP_{min}$” is developed in [16]. Their strategy involves partitioning the networks into zones and computing the power level, which may increase the overhead and lead to the degradation of the performance of the network.

The cluster architecture partitions the whole network into different groups in order to enable collaboration of sensor nodes [25]. Each cluster consists of at least two types of sensor nodes i.e., leader nodes and slave nodes. Drawbacks of choosing the cluster head statically will affect the robustness of the cluster and cannot adapt to the changing environments of the cluster [3] [9]. In the dynamic clustering architecture proposed in [3], the node closest to the target and with higher energy is chosen as the cluster head. When the target moves, the cluster heads also change. However, there is a possibility that the cluster head could move and change frequently, leading to the information transfer from one cluster head to the new cluster head, which involves costly messaging overheads. Because of this, our algorithm makes use of the features of autonomic computing to select the cluster head adaptively in a power-aware fashion.

3. Methodology

3.1. Conceptual architecture of autonomic sensor networks

The proposed architecture of an autonomic sensor network is illustrated in Figure 1. Note that although the conceptual architecture is designed as multiple layers, it does not disallow cross-layer implementation. At the bottom layer, a lightweight peer-to-peer (P2P) overlay network is being developed to support power-aware and location-aware messaging and application-level routing. This overlay network aims to enable large scales, high resilience, self-configuration and self-optimization. In particular, it accommodates stochastic redundant messaging paths, balances performance and energy consumption among sensor nodes, and strives to minimize the communication costs and elongate the lifetime of WSNs. A power-aware routing scheme is presented in 3.2, which enables the characteristics of self-configuration and self-optimization.

On the top of the P2P layer, the service discovery and coordination layer offers a generic interface to support service composition at runtime and specialized coordination mechanisms for specific applications. By self-configuration provided by the P2P overlay network layer, sensor nodes communicate with each other actively and register their services in the network. Service discovery can be used in two ways. One is to find the service for a particular event or sensed data and send the data to the service provider to process. Previous work in the field of service discovery [32] [33] [34] can be further extended and tuned for sensor networks. In contrast, the other is to load the service implementation from the service provider and compose the service locally at runtime to process unexpected data or events, which may be more energy-efficient, when the data are of high volume.

Based on these two middleware layers, the collaborative information processing algorithms and strategies can be implemented and deployed, which are driven by the requirement analysis and modeling for particular applications. Further, the results derived from the collaborative information processing layer will be analyzed and evaluated. If further processing is
required, the feedback control mechanism will be triggered to inject adjusted requirements into the lower layers in an autonomic sensor network. The feedback control mechanism can be implemented at leader nodes in a self-organized hierarchical manner, i.e., heterogeneous network which is much more cost effective than homogeneous one as shown in [18]. Self-healing can be integrated into these two layers to increase the collaborative robustness of local cluster. First, for a leader loader it should be able to recover from faults to guarantee that the current state of the target can be correctly transformed to the next cluster if it has enough energy. Otherwise, if the leader node fails, the same work should be either redone or given up. Second, the slave nodes in the cluster only need to recover them, when the performance of the cluster is below a predefined threshold.

Security issues need to be considered since many sensor nodes are deployed unattended and are prone to intrusion and masquerade. Self-protection is a critical capability to address various security issues. A special sensor called sentinel sensor can be assigned the role to monitor behaviors of the system and control access requests from end users. Trusted computing, which is proposed to enhance the security of computing systems [1], can be further employed in this autonomic sensor network architecture to protect the network from malicious attacks and unauthorized access. These sentinel nodes have the knowledge of verification. If the external access such as end user queries, collaborative requests, and data migration, cannot verified by sentinel nodes, it should be denied. Combined with self-healing, sensor nodes can be more robust and resilient.

On the top of these enabling layers are various sensor network applications. These applications involve data sensed through varied modalities such as acoustic, light, temperature and so on.

### 3.2. Power-aware routing scheme

As mention above, the energy in sensor networks is mainly consumed by communication between sensor nodes. Two ways to reduce the communications are 1) reducing the distance between two sensors and 2) reducing the communication requests. Once a cluster is formed with assigning a leader node, it should be able to configure its internal communication to collect data from slave nodes. Because of variant physical characteristics such as distance, modality, and noise model of individual sensors, data from different sensors can have various qualities. The accuracy depends on which sensor the leader node selects. On the other hand, to reduce the communication cost between leader nodes and slave nodes, leader nodes should be selected dynamically. Therefore it is critical for the local cluster to select appropriate leader and slave nodes at runtime. The proposed power-aware sensor selection method for leader node and slave node is illustrated in Figure 2.

![Figure 2. Power-aware sensor selection algorithm in local cluster](image)

Sensor nodes are deployed in clusters in such a way that there is at least one leader node to collect data and the remaining slave nodes to sense and send data. Figure 3 details the flowchart of proposed algorithm for leader node selection of phase a in Figure 2. Initially some of the nodes in the cluster can detect the target, $T$, as shown in Figure 4. If there exist several candidates of leader nodes, the one with the highest power is chosen as the leader node. The leader node does not change as the target moves until its power is below a certain threshold. Once it reaches the threshold value, the leader node automatically chooses a leader node with the highest energy level as the new one. Therefore, communication cost between the leader node and slave node can be reduced since they are around the target. In addition, such leader node selection method self-configures the cluster by choosing the highest power dynamically. If there are no leader nodes detecting the target, $T$, the leader nodes with higher energy detected by the slave nodes is chosen as the leader node. Then the selected leader
node, \( ln \), sends querying requests to the slave sensors within its range to localize the target, \( T \). After comparing the information and the communication cost among \( A \), \( B \), and \( C \), \( ln \) decides to invoke \( A \) at time \( t_1 \). Successively it is highly possible that the same node would be selected repeatedly. As a result, \( A \) would die earlier than other nodes in the local cluster, which reduces the network lifetime and may cause the network partition.

Figure 3. Leader node selection (Phase a)

The proposed power-aware slave node selection algorithm, in phase b of Figure 2, optimizes three important factors that affect the performance and lifetime of sensor networks. They are (i) information that can be transferred from the candidate sensor; (ii) communication cost; (iii) power stored by the candidate slave nodes. The selected leader node formulates them to choose the optimal sensor, which can provide satisfied data of the target and balance the energy level among all the sensors, by the combinative measurement \( \lambda_i \),

\[
\lambda_i = \alpha \phi(Z_i) - \beta \theta(j, i) + \chi \epsilon(i) \tag{1}
\]

where \( \phi(Z_i) \) can be the information utility of node \( i \); \( \theta(j, i) \) represents the communication cost between the leader node \( j \) and slave node \( i \); \( \epsilon(i) \) reflects the current energy level at node \( i \). As we can see from above equation, the value of \( \epsilon(i) \) will decrease, when node \( i \) is selected several times. Although in the successive time sequence, \( i \) still could provide better sensing data than other nodes in the cluster, to prolong the system lifetime as a whole other sub-optimal nodes should be chosen in a power-aware fashion. Consequently to select the optimal sensor based on (1), the objective is to choose the node \( i \) so that

\[
i = \arg \max_{i \in N} \lambda_i \tag{2}
\]

where \( N \) is the set of remaining sensor nodes to which the leader node sends querying requests.

We adapt the three coefficients in equation (1) at runtime because of the flexibility of sensor networks with respect to topology, noise model, movement of sensors or targets, etc. A promising solution is to make sensor networks self-optimizable, one of the key characteristics of autonomic computing. We first define system utility, \( u_{s,t} \), for a local cluster at time \( t \) as the sum of all the combinative measurements of sensor nodes,

\[
u_{s,t} = \sum_{i=1}^{N} \lambda_i \tag{3}
\]

The system utility indicates that, when the energy of the local cluster decreases, \( \chi \) should be assigned a larger value compared to \( \alpha \) and \( \beta \). In this case, the leader node will tend to select the nodes with more energy. For specific tasks, a set of thresholds and
corresponding values are predefined and stored in the cluster head.

As introduced in [14], the main function of self-optimization is to adaptively tune system parameters to keep systems performing correctly and efficiently. In our case, the leader node is able to seek ways to improve the balance of three parameters. Moreover, the proposed method also eliminates the communication costs of these parameters between candidate nodes and the leader node; then the whole bandwidth can be dedicated to transform sensing data.

4. Experimental Evaluation

The objective of the experiment is to compare the network lifetime using the proposed power-aware sensor selection scheme with that using the baseline scheme, which does not consider the energy. The experiment is based on TOSSIM [15], a dedicated simulator for TinyOS applications programmed on Berkeley motes. In the simulated scenario, initially, a certain number of sensors are deployed randomly to form a local cluster. The slave nodes are then activated to perform some processing tasks and send data to the cluster head as shown in Figure 4.

4.1. Simulation setting

In this simulation, there are 20 nodes in a local cluster; these nodes can reach the cluster head via one-hop communication (Figure 5). We assume that the initial energy of all the slave nodes and the energy consumption of transmission from each slave node to the cluster head are the same respectively. In addition, due to the limitation of TOSSIM, the sensing data from different slave nodes are assumed to have the same quality so that the effect of the first item of equation (1) is masked.

According to the proposed power-aware sensor selection method, the cluster head is able to coordinate the local cluster and activate the selected slave node. Through coordination, the head can manage the transmission power threshold; it also decides and selects an appropriate slave node to be activated in the local cluster. The cluster head collects the residual energy in slave nodes and take the average as the threshold. When a slave node attempts to send data to the cluster head, it compares its remaining energy with the threshold to decide whether to send or not. The slave nodes in the baseline scheme, however, communicate with the cluster head at random.

We run the simulation for both schemes until all the slave nodes are depleted, and compare the network lifetime at a sequence of snapshots. The comparison results are presented in the next section.

![Figure 5. Screenshot of the visualization of the cluster simulated by TinyViz. Left: sensors and the radio communication links from slave nodes to the cluster head. Right: messages transmitted in the cluster](image)

4.2. Results and discussion

Figure 6 plots the simulation results of the method with power-aware against the baseline scheme. For comparison, we sampled several percentages to indicate how many sensors in the cluster are depleted. From this figure, we observe two important results.

![Figure 6. Simulation results of a sensor cluster with the power-aware selection scheme versus the baseline scheme (X axis: the percentage of the nodes in the cluster that run out of energy; Y axis: time unit)](image)

First, at all sampling levels the proposed scheme demonstrates longer lifetime than the baseline scheme. One reason is that, if a node has less energy than the others, the cluster head does not tend to activate it in the proposed scheme. Instead, other nodes with more
power will be requested to send data to the cluster head. Therefore the lifetime of the network is extended. The baseline scheme, however, has a shorter network lifetime due to continuously consumption of a small number of nodes.

Second, the power-aware selection method can maintain the energy of the nodes in the cluster at nearly the same level. In other words, the cluster can work during its lifetime with nearly all slave nodes. The power-aware scheme keeps the survived number of sensors by balancing the residual energy of the slave nodes. From Figure 6, we note that it takes around 15 time units from “25%” to “100%” for the baseline scheme; in contrast, it takes only around 7 time units for the power-aware scheme. In the final phase of the cluster, the system using the baseline scheme works with less number of slave nodes, which degrades the performance of the sensor network due to connectivity and coverage.

5. Conclusion

Wireless sensor networks promise to revolutionize our lives by embedding and networking sensors pervasively, while the challenge is how to design sensor networks to achieve high performance and long lifetime. In this paper, we presented a novel conceptual architecture, i.e., autonomic sensor network, to equip the sensor network with the self-managing capability. Specifically, we design the system layer by layer and investigate the benefits by integrating the self-manageability.

To realize self-optimization and self-configuration aspects of an autonomic sensor network, a novel power-aware sensor selection scheme is developed. Local clusters are self-configurable to choose the leader node; information from slave nodes is collected and processed collaboratively. We confirm by simulation that the proposed scheme prolongs the network lifetime and balances the energy level in the local cluster. Further, self-optimization of adaptation parameters is investigated. Evaluations of the proposed autonomic sensor network system middleware are ongoing and will be reported in our future publications.

References


