Sensor Placement with Multiple Objectives for Structural Health Monitoring

MD ZAKIRUL ALAM BHUIYAN and GUOJUN WANG, Central South University
JIANNONG CAO, The Hong Kong Polytechnic University
JIE WU, Temple University

Structural health monitoring (SHM) refers to the process of implementing a damage detection and characterization strategy for engineering structures. Its objective is to monitor the integrity of structures and detect and pinpoint the locations of possible damages. Although wired network systems still dominate in SHM applications, it is commonly believed that wireless sensor network (WSN) systems will be deployed for SHM in the near future, due to their intrinsic advantages. However, the constraints (e.g., communication, fault tolerance, energy) of WSNs must be considered before their deployment on structures. In this article, we study the methodology of sensor placement optimization for WSN-based SHM. Sensor placement plays a vital role in SHM applications, where sensor nodes are placed on critical locations that are of civil/structural engineering importance. We design a three-phase sensor placement approach, named TPSP, aiming to achieve the following objectives: finding a high-quality placement for a given set of sensors that satisfies the engineering requirements, ensuring communication efficiency and reliability and low placement complexity, and reducing the probability of failures in a WSN. Along with the sensor placement, we enable sensor nodes to develop "connectivity trees" in such a way that maintaining structural health state and network connectivity, for example, in case of a sensor fault, can be done in a distributed manner. The trees are constructed once (unlike dynamic clusters or trees) and do not incur additional communication costs for the WSN. We optimize the performance of TPSP by considering multiple objectives: low communication cost, fault tolerance, and lifetime prolongation. We validate the effectiveness and performance of TPSP through both simulations using real datasets and a proof-of-concept system on a physical structure.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, sensor placement, communication efficiency, fault tolerance, connectivity, lifetime, structural health monitoring

68

A preliminary version of this work has appeared [Bhuiyan et al. 2012c] in *Proceedings of the 14th IEEE International Conference on High-Performance Computing and Communications (HPCC12)*. This work is supported by the National Natural Science Foundation of China under grant nos. 61272151 and 61272496, the International Science & Technology Cooperation of China under grant no. 2013DFB10070, and the "Mobile Health" Ministry of Education – China Mobile Joint Laboratory (MOE-DST no. [2012]311). This work is also supported by HKRGC under GRF grant PolyU5106/11E and HKPolyU Niche Area Fund 1-BB6C, and by NSF under grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.
Author's addresses: M. Z. A. Bhuiyan, School of Information Science and Engineering, Central South University, Changsha, China, 410083 and Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong; G. Wang (corresponding author), School of Information Science and Engineering, Central South University, Changsha, China, 410083; email: csqjwang@gmail.com; J. Cao, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong; J. Wu, Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122.

DOI: http://dx.doi.org/10.1145/2533669

ACM Reference Format:
Md Zakirul Alam Bhuiyan, Guojun Wang, Jiannong Cao, and Jie Wu. 2014. Sensor placement with multiple objectives for structural health monitoring. ACM Trans. Sensor Netw. 10, 4, Article 68 (June 2014), 45 pages.
DOI: http://dx.doi.org/10.1145/2533669

1. INTRODUCTION

Wireless sensor networks (WSNs) have been deployed in a wide variety of applications related to public safety, such as military surveillance, intrusion detection, and tracking applications [Akyildiz et al. 2002; Wu 2005; Wang et al. 2010; Bhuiyan et al. 2013b]. Structural health monitoring (SHM) is also becoming a prominent application of WSNs in the interest of public safety, for instance, smart buildings, bridges, smart aircrafts, dams, etc. These consist of *physical* components (see Figure 1). Current implementation costs and time for a wired sensor network are high, making the realization of a large-scale network prohibitively expensive. Compared to the wired network-based systems, WSN-based SHM systems with optimized control techniques are believed to have the potential to offer not only a healthy and comfortable environment, but also lower maintenance costs and time. Such a system is a typical example of a complex cyber-physical system (CPS) [Hackmann et al. 2013]. However, the nature of SHM poses many new challenges to resource-constrained WSNs.

First, sensor placement plays a key role in SHM, according to the civil and structural engineering (CSE for short) domains [Beygzadeh et al. 2013; Li et al. 2010; Yi et al. 2011]. Sensors have to be placed at critical locations that are of the CSE's importance, and provide the health state accurately. These placement methods require significant domain knowledge along with SHM complexity. Conversely, sensor placement in generic WSN applications is often assumed random, uniform, on grids, tree, polygon, etc. With these sensor placement methods, effective SHM may not be possible because the spatial information to describe the dynamic behavior of a structure or sensitivity of events of interest (e.g., damage) is not sufficient at many locations.

Second, wired sensor networks are usually deployed for SHM by the CSE domains. Only a few justify the use of WSNs, where sensors are often required to transmit data to a central server (or a *sink*) [Ceriotti et al. 2009; Araujo et al. 2012; Hackmann et al. 2012; Li et al. 2013; Linderman et al. 2010; Farrar and Worden 2012]. By using either single-hop or multihop communication techniques, data transmission over a large structure is extremely difficult and costly. These techniques bear the common problem of relying on the sink (bottleneck).

Third, on the one hand, SHM requires vibration measurement data at a high rate and demands efficient delivery for a large amount of data, which must be divided into a large number of packets [Kim et al. 2007; Wijetunge et al. 2009]. Each sample data point is typically 2 to 4 bytes, and sampling rate can be from hundreds to thousands of Hertz. On the other hand, the wireless communication links are very volatile and prone to failure, due to channel effects, interference, sensor faults, etc. Thus, on the CSE-driven deployment, it is difficult to balance the wireless data communication load.

Fourth, existing WSN deployments for SHM systems primarily focus on cyber aspects (like data acquisition and communication) or on structural physical aspects (like damage detection or modal analysis). But the deployments do not focus on the integration of both types of aspects (called cyber-physical system (CPS) aspects) [Peckens and Lynch 2013; Anshuman et al. 2013; Bocca et al. 2011; Farrar and Worden 2012; Yi et al. 2011; Araujo et al. 2012; Bhuiyan et al. 2013a]. This isolation may result in suboptimal CPS solutions.

A representative example. It can be best observed that, commonly, “no change (e.g., damage)” occurs in a structure. Regarding this in a resource-constrained WSN, the large volume of data really does not always need to be transmitted to the sink. Instead, a simplified decision transmission may be interesting in the case of a "no change" state. However, if a change has occurred in the structure, in addition to transmitting a decision on the possible change, a node may need to transmit all of its collected data towards an upstream node or the sink upon request. The nodes may have additional interactions between them. For example, they need to communicate to the neighboring nodes in the region of the change and to further analyze the data for ensuing the state of the change. This indicates that a change in the physical structural system results in extensive communication and computation in the WSN system, especially between the nodes in the region around the change.

Therefore, the integration between both systems is a CPS. In such a system: (i) a distributed decision maker, and (ii) finding the regions of interest, are essential so as to reduce the volume of data in the case of "no damage." If there is damage at a part/section/span/region of the structure (say, in a *substructure*; refer to Figure 1), a group of sensors (say, a *subnetwork*) around the damage should operate for a prolonged time. Sensors in the other subnetworks (or in the other regions) can sleep to extend their lifetime.

Motivated by the preceding limitations and requirements, in this article, we study the sensor placement problem in a heterogeneous WSN and design a three-phase sensor placement (TPSP) approach. TPSP is composed of two types of sensor nodes: low-end nodes (LNs) and high-end nodes (HNs). The LNs are resource constrained (e.g., limited battery lifetime and communication abilities), while the HNs are resource rich (e.g., long battery lifetime and long-distance communication abilities). A mixed deployment of these nodes can provide a balance of performance and cost in WSNs.

In order to achieve an accurate monitoring decision, the placement of such HNs and LNs is carried out efficiently and effectively. For example, in the case that there is *damage* somewhere in a structure, LNs process the collected data and transmit an extremity of the damage state to the HNs. Besides an LN's tasks, an HN also aggregates all of the states. Along with the LNs' placement, we enable each HN to maintain a connectivity tree, called "HN-LN" tree wherein the HN can perform appropriate actions (sending a refined decision, e.g., *alert*, tackling routing inconsistencies, sensor faults, maintaining connectivity, etc.) in a distributed manner.

The optimality criterion used as a *location-quality indicator* is the optimization of the determinant of the Fisher information matrix (FIM). This is a standard metric to identify the placement quality according to the CSE domains [Kammer 1990; Meo and Zumpano 2005; Yi et al. 2011]. Utilizing the location quality, we deal with the placement problem with a large finite set of feasible M locations in a structure. This eliminates the least effective locations until a given set of $N(<M)$ sensors is left.

We perform the sensor placement in three phases. In the first two phases, we place a set of HNs and a set of LNs, respectively. The number of HNs can be limited while that of LNs can be large, relying on the scale of the WSN and the structure. However, the WSN formed by the HNs and LNs lacks some requirements, such as low communication load, network connectivity, and fault-tolerance ability. Because traditional placements in CSE do not focus on these requirements, CSE still prefers the use of wired networks for SHM that are not usually prone to fault/failure, even though in many cases there is no need to tackle a fault.

We improve these situations by placing a set of redundant nodes (RNs) whose functionality is similar to the LNs. This is performed by finding possible vulnerable points in the WSN (e.g., articulation point or junction point) out of the remaining locations from M locations. As a result, multiple objectives, such as low communication, a specific level (e.g., k-1) of fault tolerance, and prolonged lifetime, can be achieved.

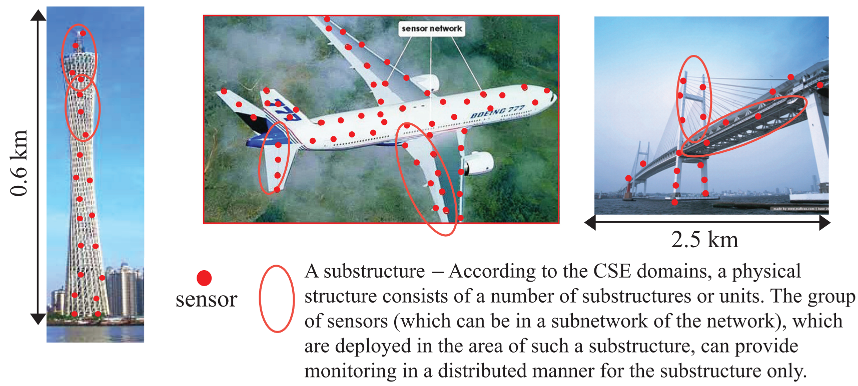


Fig. 1. Example CPSs in which sensors can be placed across the structures to perform active monitoring and control in a distributed manner.

In summary, this article makes a number of major contributions as follows.

—We formulate a sensor placement problem as a multiple-objectives optimization problem for a WSN-based SHM. This is a difficult task, as it requires multi-domain knowledge to satisfy the domain's specific requirements. We design the TPSP approach to address the problem.
—We propose a set of algorithms, including a new notion of a priority-based redundant sensor placement (PRSP) method, to improve the WSN's performance.
—To achieve the monitoring in a distributed manner, we develop an `HN-LN` tree along with the sensor placement. It does not incur additional costs for the WSN. Network maintenance (e.g., in the case of sensor fault) is also achieved by such a tree.
—We evaluate our TPSP approach via simulations using real datasets collected by a SHM system (a high-rise building monitoring project of Hong Kong PolyU)[1]. Also, we implement a proof-of-concept CPS using the Imote2 platforms and TinyOS on a civil structure. The results, compared with existing work, validate the efficiency and effectiveness of the TPSP approach.

This article is an extended version of a preliminary work [Bhuiyan et al. 2012c]. The extension includes several aspects: (i) a detailed analysis of system models, constraints, and the algorithms with complexities is carried out; (ii) a new algorithm (Algorithm 5) for fault tolerance and a method for improving difficulties in WSN-based SHM are presented; (iii) clarifications on the duty cycle for nodes, simulation, and deployment setup and related work are made; (iv) further evaluation and comparisons between this approach and existing approaches are conducted.

The layout of this article is as follows. Section 2 reviews related work and provides some background of SHM. Section 3 presents the problem formulation and models. In Section 4, we develop three-phase sensor placement (TPSP) algorithms. Section 5 analyzes the improvement on both WSN and SHM system aspects and fault tolerance. We evaluate our approach via simulations in Section 6 and a real implementation in Section 7. Finally, Section 8 concludes this article and highlights our future work.

2. RELATED WORK

In this section, we relate our approach to existing work.

[1]Structural health monitoring for Guangzhou new TV tower, http://www.cse.polyu.edu.hk/benchmark/.

2.1. Existing WNS-Based SHM Approaches

Wireless sensors, with their embedded computational and communication capabilities, offer new opportunities for SHM. Over the past several years, the CSE domains gradually have targeted the implementation of analytical methods to detect and quantify structural damage by using reliable sensing technologies [Farrar and Worden 2012; Linderman et al. 2010; Rice and Spencer 2009; Bocca et al. 2011; Whelan and Hietbrink 2012; Peckens and Lynch 2013; Araujo et al. 2012]. While WSNs are gradually receiving attention from the CSE as an attractive tool, many engineering applications (e.g., SHM) are receiving attention from the computer science domain. Some research efforts on this overlapping area have been presented in the literature [Jindal and Liu 2012; Li et al. 2010; Liu et al. 2011b; Hackmann et al. 2013; Peckens and Lynch 2013; Bhuiyan et al. 2012b; Ceriotti et al. 2009]. A state-of-the-art technique verified on the Golden Gate Bridge (GGB) and applications of WSNs for SHM can be found in Kim et al. [2007], Ceriotti et al. [2009], and Wijetunge et al. [2009].

2.2. Hierarchies in WSNs for SHM Approaches

Traditional wire-based network deployments for SHM rely on flat network architectures [Ni et al. 2008]. The decision on structural state is made offline. Many existing WSN approaches use the flat architectures [Bocca et al. 2011; Whelan and Hietbrink 2012; Peckens and Lynch 2013; Araujo et al. 2012; Li et al. 2013, 2010; Kim et al. 2007]. The way of raw data transmitting and processing at the sink makes it practically difficult to achieve the high quality of monitoring, because of WSN constraints. Like the generic WSN applications (e.g., target/event detection and environmental monitoring), there is also a growing demand for an increase in transmission bandwidths in WSN deployments with large networks. They require distributed aggregation of many data records.

Particularly, online SHM based on WSNs can be a promising technique for monitoring the health state of structures. There are a number of WSN approaches for SHM from both CSE and computer science domains that have assumed a hierarchical WSN architecture for structural monitoring in recent years [Jindal and Liu 2012; Liu et al. 2011b; Sima et al. 2011; Nie and Li 2011; Bhuiyan et al. 2012b; Hackmann et al. 2013; Liu et al. 2011a]. However, none of these approaches focuses on how to deploy a WSN deployment in a hierarchical manner.

Cyber-Physical Systems (CPS) have recently been proposed by Li et al. [2013] and Hackmann et al. [2013], considering both the constraints of the underlying WSNs and the SHM requirements, where a distributed damage detection algorithm is suggested under a hierarchical cluster-based WSN [Hackmann et al. 2013]. Although it offers a trade-off between the computation and communication capacities, it does not reveal the details, such as connectivity, or coverage in the hierarchical architecture. A pure hierarchical Cluster-based SHM (C-SHM for short) considers a fundamental problem in SHM: mode shape analysis (see Definition 3.2) in clusters [Liu et al. 2011b]. In each cluster, the vibration characteristics are identified and then assembled together.

However, the heuristic-based clustering approach is used in the preceding approaches, which may not fit a real SHM. Although the clustering in the C-SHM and Hackmann et al. [2013] needs to meet extra requirements of modal analysis, the quality of SHM may be affected because the modal analysis can be different at the same cluster at different times. The cluster size should be optimized to minimize the total energy consumption. Moreover, the system performance through the hierarchical architecture of both C-SHM and Hackmann et al. [2013] in terms of connectivity, communication, lifetime, etc., is not discussed. Each cluster of nodes cannot provide an analysis on a state of the cluster region independently.

2.3. Existing Placement Approaches and their Challenges

Generally, a large number of sensors are involved in monitoring a large structure. Numerous engineering methods for wired systems have been advanced for the optimal sensor placement problem. These methods are widely reported in the engineering literature, such as the effective independence method (EFI), kinetic energy method (KEM), and genetic algorithm [Kammer 1990; Meo and Zumpano 2005; Yi et al. 2011; Beygzadeh et al. 2013]. When deploying a WSN for SHM, these methods pose many challenges to the WSN, allowing for constraints in WSNs, for example, wireless bandwidth, communication range and load, fault tolerance, energy, etc. Conversely, although a lot of existing work is dedicated to WSN systems in different aspects [Wijetunge et al. 2009; Araujo et al. 2012; Hackmann et al. 2013; Rice and Spencer 2009], they do not show the methodology of the WSN deployment on structures. Even then, the methods of optimal WSN deployment are more essential if one intends to deploy a minimum number of sensors.

2.4. Difficulties in Applying Generic WSN Placement Approaches for SHM

In generic WSN deployment for monitoring an event (e.g., object or target), sensor placement is often carried out randomly, uniformly, and so on. With these placements, each sensor node detects an event by comparing the received energy, in terms of light, acoustic, temperature, etc., emitted by the event to a threshold. Each sensor normally requires a lightweight computation (in many cases, 0/1 decision for the event detection). There are plenty of deployment approaches proposed by computer science researchers focusing on this research area [Chang et al. 2011; Wang et al. 2008; Kashyap et al. 2011; Krause et al. 2011; Xue et al. 2012; Xu et al. 2010; Bredin et al. 2010].

In contrast to these applications, it can be seen that WSN deployment for monitoring a structural event (e.g., damage, crack, etc.) is not as straightforward as in other applications. Detection of structural damage in SHM is performed through vibration and strain characteristics. With the generic WSN deployment methods, effective SHM may not be possible because the spatial information to describe the dynamic behavior of a structure or sensitivity of an event (damage) is not sufficient at many locations, where monitoring damage is due largely to structural location sensitivity. For example, existing sensor placement with grids, or at intersection points, may not be suitable for SHM. The intersection points of cells in a grid cannot be considered as the candidate sensor locations regarding structural properties.

Relay nodes are placed anywhere, or by using random or uniform deployment methods in WSNs, for instance, Xu et al. [2010], Xue et al. [2012], Bredin et al. [2010], and Cheng et al. [2004]. A relay node placement strategy is proposed by Xu et al. [2010] to improve a uniform deployment-based solution. This solution for data collection is a state-of-the-art approach proposed by computer scientists. It discovers that the uniform deployment strategy is inefficient from an energy perspective. This is due to the biased energy consumption rate phenomenon in both single-hop and multihop heterogeneous WSN cases. It then uses a random deployment strategy for relay nodes in the uniformly deployed WSN. In practice, there are certain restrictions (with respect to the location quality) on such a method of placement in SHM.

A closely related approach, pSPIEL [Krause et al. 2011], points out the intractability of the sensor placement problem, which incurs low communication cost between the sensors. The pSPIEL presents a polynomial-time, data-driven algorithm using nonparametric probabilistic models called Gaussian processes, both for the spatial phenomena of interest and for the spatial variability of link qualities. This allows it to estimate the predictive energy and the communication cost of unsensed locations. In pSPIEL, entropy is used to measure the quality of the data collected, and the placement is gradually refined by the gaussian process. The meaning of "informative" in pSPIEL is thus very general.

In contrast to pSPIEL, the mathematical model of a sensor placement derived from the civil engineering domain provides the information about structural geometry. For example, damage detection through the structural mode shape depends on the geometry of a physical structure. We think that the high-quality locations found through the sensor placement algorithms in pSPIEL may be used in various applications in a general sense, but they may not provide high-enough quality for SHM.

2.5. SHM Domain-Specific Sensor Placement and Our Approach

Some of the approaches from the engineering domain discussed before justify whether the deployed WSN-based SHM system can replicate the data collection functionality of the original wire-based counterpart. Many times, powerful wireless sensor nodes are deployed to accomplish tasks that could have been achieved by more cost-effective counterparts through system optimization. These approaches still have difficulties in handling the constraints of WSNs. They are also not concerned with sensor faults, communication faults, routing inconsistency, etc, leaving opportunities in this multidomain research area, such as deploying a WSN for SHM, bearing distributed monitoring in the WSN that can incorporate both computer science and CSE aspects.

In the event of a typhoon, earthquake, or damage or crack in a structure, to ensure the safety during the construction of (and the operational performance of) the GNTVT structure, a long-term SHM system has been implemented. A deployment method called SPEM [Li et al. 2010] nicely explains CSE requirements and is verified on the GNTVT. It adjusts the quality of sensor locations to better fit WSN requirements. However, such an adjustment (with a reduction of location quality) may lose some optimal locations. Also, it is fully centralized and does not have any fault tolerance support. It does not handle communication errors and load. Since all the sensors involve transmitting a large volume of data to the sink, they may fail during operation.

Our proposed three-phase sensor placement (TPSP) approach significantly differs from existing approaches. We attempt to satisfy CSE requirements, together with WSN constraints. To achieve the health state in a distributed manner in a WSN with multiple objectives, including fault tolerance, we employ the CSE's method in three phases in TPSP. Through these phases, a subnetwork deployed in a substructure can provide a health state of the substructure independently. TPSP does not alter the parameters of the structural properties. It demonstrates that it is effective in solving the optimal wireless sensor placement problem, with performance results similar to the engineering methods, with lower computational iterations.

3. PROBLEM FORMULATION AND MODELS

As the first work to deploy a multiple-objective WSN for SHM applications, we first describe preliminaries, including some necessary definitions. Then, we explain WSN system models, a sensor placement model, and associated constraints. Finally, we outline our problem.

3.1. Preliminaries

Definition 3.1 (*Finite-Element Model (FEM)*). Finite-element model is a computer-based numerical model for calculating the behavior and strength of structural mechanics, such as vibration and displacement. Via FEM, a complex structural model is simplified by breaking it down into small elements. These elements are blocks that contain the information of the entire property of the structure.

Definition 3.2 (*Mode Shape:* Φ). Each type of mechanical structure has a specific pattern of vibration at a specific frequency, called mode shape. More specifically, it basically shows how a structure will vibrate and in what pattern. Φ is the matrix of FEM target mode shapes, for instance, Mode1 (or Φ_1) : $\{2.56, 7.45, 10.56, 6.34\}$Hz.

Definition 3.3 (*FIM: Q*). A Fisher Information Matrix (FIM) is used to calculate the placement quality indicator (E_j). In CSE, the FIM determinant $|Q|$ can be given by a location assignment which is a standard metric to specify E_j.

Definition 3.4 (*Damage*). Damage is a significant change to the geometric properties of a structural system, such as changes to captured frequencies and mode shapes.

These definitions are used for sensor placement in this article, which may help with understanding some properties adopted by CSE domains. We provide the following assumptions and notations before the problem statement.

—Consider that a structure F is given for monitoring. A set P of N heterogeneous sensors will be placed by finding locations on F such that any changes related to F's health can be accurately identified.
—The set P includes a set H of n_h high-end nodes (HNs), a set L of n_l low-end nodes (LNs), and a set R of n_r redundant nodes (RNs). n_l can be plenty, but n_h and n_r are limited. Thus $P = H \cup L \cup R$, $|H| < |R| < |L|$, $|P| = N$, and $N < M$. Here, N is the total number of sensors and M is the number of feasible locations on F.
—The HNs should collect data from LNs and aggregate the refined data sent by the LNs to make a distributed decision. Basically, LNs and RNs have the same functionalities in this work, but they are placed in different phases so as to improve the network performance. Sometimes we refer to both types of low-end sensor nodes as LNs.
—The candidate sensor locations can be assigned, $S = \{s_1, s_2, s_3, \ldots, s_{|H|+|L|+|R|}\}$, from a set of M feasible locations for the set P of N sensors. The sink can be located at a suitable location s_0. Each jth sensor location has a contribution to the location quality E_j (see Definition 3.3).

3.2. Communication Model

We assume that LNs have a communication range $0 < \tau \leq R_c$, where τ is adjustable. Using R_c, an LN communicates to neighboring LNs or an HN. However, an HN has two ranges: R_c to communicate with its LNs; R_h to communicate directly with its neighboring HNs or to the sink. Here, $R_h \geq R_c$. We adopt these two types to reduce energy consumption for communication. We calculate $C.cost$ as the communication cost in the WSN (described in Section 5.2).

Let $d_{u,v}$ be the Euclidean distance between any two LNs, or an LN and an HN. That is, any two sensors u and v can communicate directly if and only if $d_{u,v} \leq R_c$, and there exists at least k paths from any LN to at least an HN. In other words, a node u can communicate directly with another node w, which could be an LN or a direct HN if and only if $d_{u,v} \leq R_c$. An HN u can communicate directly with v (which can be another HN or the sink) if and only if $d_{u,v} \leq R_h$.

Besides the energy consumption metric (modeled in Section 3.3) for communication in the network, an important communication metric is the quality of a wireless link. We attempt to achieve communication efficiency and link reliability in two stages: during sensor placement and during network runtime.

In the first stage, we calculate $Q_{u,v}$ as the probability of link quality, mainly based on the expected number of retransmissions between locations of any two sensors u and v, as suggested by Krause et al. [2011]). In the second stage, we consider that some of the routing paths through the tree may be altered during monitoring operations. We apply the technique of datapath validation [Gnawali et al. 2006], which enables routing layers to remain efficient and reliable in highly dynamic topologies. See Appendix A for a detailed description of these two stages.

Following the aforesaid policies, the HNs, the LNs, the RNs, and the sink, together with the values of R_h and R_c, collectively generate a heterogeneous network. We describe the following constraints for generating the WSN.

C1 (*Connectivity Constraint*). Given HNs, LNs, and RNs to be placed to form a heterogeneous WSN, where HNs use both adjustable R_h and R_c and all of the LNs adjust τ up to R_c, determine the transmission power level τ of each sensor such that:

—k-node HN connectivity: there exist k-node disjoint paths from every LN to other LNs and at least one HN, and every HN to the sink;
—the maximum transmission power level is reduced, $MAXp_\tau|\tau = 1, 2, \ldots = minimum$; and
—the WSN can tolerate the failure of up to $k - 1$ nodes.

C2 (*Transmission Load Constraint*). The maximum distance $d(> d_{u,v})$ from any LN to an HN and an LN to another LN should be minimized. As a result, the load of long-distance transmissions from an LN to any neighboring LNs and an LN to its corresponding HN may alter.

C3 (*Data Delivery Constraint*). Decisions about the structural health state made by each LN are gathered at one or more HN(s). The final (distributed) decision made by an HN should be quickly forwarded to the sink only when there is a "damage" state otherwise, a normal acknowledgment is forwarded. Thus, each sensor must periodically report its decision, and data delivery must be fault-tolerant to the sensor failure of up to a specified level (e.g., $k - 1$).

3.3. Energy Model

For the estimation of the required transmission energy, we follow a standard transmission model Cheng et al. [2004; Wang et al. [2006]. Such a model assumes that the energy per bit for transmission over a wireless link is a function of the distance between a transmitter and a receiver. Let E and T be the maximum energy consumption and the system lifetime, respectively. A node u is assumed to generate data at a rate x_{uv} during its lifetime. Similar to the model Cheng et al. [2004], the energy required per unit of time from a node u to a node v is determined by

$$E^t_{uv} = p^t_{uv} \cdot x_{uv}, \tag{1}$$

where p^t_{uv} is the required energy for transmitting one unit of data at time t, which can be modeled as

$$p^t_{uv} = \alpha + \beta \cdot d^{\kappa}_{uv}, \tag{2}$$

where α and β are nonnegative constants, and κ is the path-loss exponent parameter in $\{2, 6\}$. These parameters depend on the structural environment (i.e., bridge, building, or others). This *ideal power model* in (2) has been widely adopted to study various theoretical aspects of WSNs [Cheng et al. 2004; Olariu and Stojmenovic 2006; Singh et al. 1998]. We focus on the Per-packet cost minimization model (PPCM), in which the data is sent on a path that minimizes the total energy consumption to deliver the data [Singh et al. 1998]. We define the neighbors of node u as $N(u) = v \in P|d_{u,v} \leq R_c$ or R_h.

In the preceding, we discuss the energy consumption for data transmission. We assume that a sensor node normally has different states of operation, namely working, sleeping, and idling [Wang et al. 2006]. In the idle state, the node consumes a significant amount of energy. We use the rate of energy consumption for a node's idle state modeled in Wang et al. [2006]. There are other sources of energy consumption that we consider

Table I. E_j Values Obtained from Different Candidate Locations

Candidate location number	E_j
332	0.9234
448	1.0000
45	0.6401
221	0.9981
96	0.2312

when estimating the total energy consumption in the WSN. Examples include network maintenance (fault tolerance, datapath validation), any retransmission due to packet loss, as well as clock synchronization.

Definition 3.5 (*T: Lifetime*). This is the elapsed time from the launch of a WSN until the instant that an HN fails to receive data from an LN or an HN due to energy depletion.

For our network setting, Definition 3.5 is equivalent to the minimum lifetime of the sensors [Cheng et al. 2004], that is,

$$\mathbb{E}[T] = \mathbb{E}\left[\min_u(T_u)\right], \tag{3}$$

where T_u is the lifetime of sensor u.

To ensure a maximized system lifetime besides ensuring the quality of monitoring, an appropriate sensor node duty cycling should be employed in which the radios can go to sleep and wake up periodically. A discussion on the duty-cycle setting in this approach can be found in Appendix B.

3.4. Sensor Placement Model

Optimal sensor placement maximizes the possibility of identifying structural health states. Based on the assumptions from CSE, the health state of a structure, such as damage, depends strongly on the location-quality indicator (E_j). Computational approaches are used to place sensors at the optimal locations.

Our aim is to satisfy aspects of both CSE and computer science domains for a WSN-based SHM. This is why we follow the widely accepted sensor placement method from CSE, effective independence method (EIM). The mode shape Φ is calibrated for finding locations. The EIM optimizes and selects a set of target modes (e.g., N) for identification based on the FEM analysis. An initial candidate set of N sensor locations can be selected from a feasible set of M locations.

The quality indicator related to the jth sensor location is within the range $0.0 \leq E_j \leq 1$. The mathematical formulation to obtain E_j via EIM can be found in Appendix C. On the Matlab platform, 110 candidate sensor locations, based on datasets from GNTVT (see Footnote 1 in Section 1)were ranked using E_j. For example, five candidate locations with their location quality are listed in Table I. $E_j = 0.0$ indicates a location having no contribution to the measurement data matrix. Such a sensor location is removed from the candidate set without impacting Q. $E_j = 1.0$ indicates a location that highly contributes to the measurement data matrix and should not be removed from the candidate set.

It is sensible that the larger the contribution to E_j, the better the location quality (see Table I). A high-quality location can be set by the quality indicator, for example, $E_j \geq 0.7$, $E_j \geq 0.5$, etc., depending on the end SHM user and the number of sensors available. However, one of our important purposes is to improve network performance for the SHM system, thus the sensor placement is subject to the following constraints.

C4 (Location Assignment Constraint). Two HNs or LNs should not be placed at the same location, but an RN can be placed at an LN's location as required. For any two HNs, u and v, $d_{uv} \geq R_s$, R_s is the sensing range.

C5 (Structural Constraint). In a real-world SHM, the selected optimal locations on a structure for placing sensors may be inaccessible. Also, wireless sensor communication is less reliable ($Q_{u,v}$ is low) in structural environments due to several reasons.

—*Physical structural model*. There is the proximity to the bearing load zone (e.g., in a bridge structure or in an industrial machine), or obstacles (wall, pillars, piers, etc.), or environmental effects (e.g., heavy wind).
—*Irregular sensing field*. There is the irregular sensing field constraint, for example, not a square or circle shape on a building, aircraft, or other structure [Linderman et al. 2010].
—*Location*. The location is a poor/vulnerable point for wireless sensor communication. The link quality may also vary from one location to another or one substructure to another or one structural environment to another.
—*Environmental Interferences*. The environmental interference is very high.

3.5. Problem Statement

A set P of N sensors is given to be placed to form a WSN for monitoring F by finding locations $S = s_1, s_2, \ldots, s_N$ with maximum E_j out of a finite set of M candidate locations of F, such that $C.cost$ is minimized, the WSN guarantees to tolerate the failure of up to $k - 1$ sensors, and T is maximized, subject to the C1 to C5 constraints.

4. THREE-PHASE SENSOR PLACEMENT (TPSP)

In this section, we present several algorithms to design a heterogeneous WSN. At first, we place HNs and LNs in the first two phases, respectively. Then, they are connected and organized into groups. Finally, we provide the third phase for RNs' placement.

4.1. The First Phase: HNs Placement

Data transmission from sensor nodes to the sink often incurs significant energy consumption. This critically affects T, especially in SHM application. Suppose that we are given a substructure (e.g., a long span of a bridge, a number of floors of a high-rise building, or a large section of an aircraft) of a large civil structure for monitoring. Neither a strategy of long-range one-hop data transmission nor short-range hop-by-hop communication is cost efficient. We generalize and reduce energy consumption by deploying several HNs in terms of single-to-multihop communication.

As can be seen in generic WSN deployment, relay nodes, leader nodes, cluster head, high-end, or similar types of nodes are deployed in the last phase, that is, after the deployment of a given set of sensors [Xue et al. 2012; Xu et al. 2010; Wang et al. 2011; Kashyap et al. 2011]. In this work, we attempt to place the HNs in the first phase, due to several benefits for SHM. For instance:

—we determine the optimal, also suitable, locations for the HNs according to our communication model, wherein we take into account the high communication link quality between any two HNs in such a way that enables us to achieve monitoring decisions from each substructure independently (thus, the HNs placed at such locations should provide a balance of cost in the network);
—every HN also has a sensing task wherein, since they are placed at the best locations of a substructure, the ability of health monitoring is enhanced; and
—HNs are expected to work longer than LNs.

4.1.1. Substructure-Oriented Subnetwork Deployment. We can gain the previous benefits through a substructure-oriented subnetwork deployment. As illustrated in Figure 1, a large physical structure consists of a number of substructures. In other words, a structure can be divided into a number of substructures based on different sections. After deployment, sensor nodes in a WSN can be organized into groups (or subnetworks) in such a way (in a hierarchical manner) that each subnetwork can provide localized monitoring results for a substructure independently.

However, such substructures are usually identified by wired sensor networks (having constant power supply). This cannot be possible by the wireless sensors, as the identification requires a huge amount of high-complexity computation and communication. The optimal sensor locations in such substructures can also be identified by the wired sensors. For example, see the identification of substructures (sections/regions/subspaces) of a structure using wired sensors and the group of wired sensors connected to each other and to the sink [Ni et al. 2009; Xing and Mita 2012; Weng et al. 2011; DeVore 2013].

Considering severe constraints in WSNs, we want to find localized decisions on the health state of each substructure independently by the subnetwork. To achieve this, it can be better if we can still organize the nodes according to the physical substructure orientation. In recent years, a number of WSN approaches from both CSE and computer science domains have employed hierarchical WSN architectures for structural monitoring [Jindal and Liu 2012; Liu et al. 2011b; Sima et al. 2011; Nie and Li 2011; Bhuiyan et al. 2012b; Hackmann et al. 2013]. However, these approaches do not figure out how to get such a WSN deployed for SHM. Thus, we have to find a way that can create subnetworks of a WSN without identifying substructures.

4.1.2. Traditional Approaches to Find Subnetworks. On the one hand, one may wish to find a heuristic-based sensor organization [Liu et al. 2011b; Jindal and Liu 2012; Hackmann et al. 2013]. We think that such a network may not provide the coverage of the substructure in practice and may not come as a result of optimal sensor placements. The dynamic sensor organization incurs a significant amount of communication cost.

On the other hand, one may wish to perform sensor placement in a greedy fashion (similar to procedures in EIM): starting with one sensor node, each node is placed, one at a time, so that it optimally compliments the existing fixed arrangement. It allows for the finding of a subset of locations with high quality, in other words, to simply pick sensor locations in sequence with high-quality indicators, irrespective of the communication cost and link reliability in the network, and also to the substructure-oriented monitoring.

However, such a simple greedy solution for sensor placement performs arbitrarily worse than a global optimum solution for sensor placement, in practice. Unfortunately, the optimality of the simple greedy algorithm only holds when we do not take into account the communication cost and reliability. It also holds when we do not generalize to the strong connectivity between the nodes. Since EIM is basically designed for wired sensor deployment, there are usually no such difficulties.

One may think of a global optimum solution that may be an alternative for sensor placement to the simple greedy strategy when deploying all of the sensors at one deployment time. However, such a solution may only be used for a new system deployment for a structure at one deployment (with no support for further improvements). It does not suit the substructure-oriented monitoring. Also, the solution does not work well to counter physical obstacles or network communication holes in diverse structural environments.

In contrast, we present algorithms (Algorithm 1 for HNs' placement and Algorithm 3 for RNs' placement) that have a modification on the simple greedy algorithm. They are useful when augmenting an existing system deployed with additional sensors, or when

ALGORITHM 1: Optimal Location Finding for HNs

```
Input: Given a set H(⊂ P) of n_h HNs, M candidate locations.
Output: Location assignment for the set H of n_h HNs.
Step 1: generate Φ^m    // using Eqn. (6), as derived in the Appendix C;
        compute initial data matrix X with n_h rows;
Step 2: for i = M to n_h + 1 do
            compute E_j;
            M ← sort locations of M, according to E_j;
        end
Step 3: select jth location, j = 1;
        s_j ← j;    //set a location
        place an HN at s_j;
        for j = 1 to n_h − 1 do
            for i = 1 to M − 1 do
                if there is any HN's location with high Q_{u,v} then
                    i = i + 1;
                else
                    s_j ← i;
                    place j at s_j;    //place the next HN
                end
            end
        end
    return placement of n_h;
```

one wishes to deploy more sensors to enhance the system performance. They are also useful for providing communication cost balance and reliability into the network.

4.1.3. The HNs' Placement Algorithm. For the HNs' placement (through Algorithm 1), we can determine the diameter of a given structure and all of its substructures. The diameter is used to determine the distance between two substructures. On the one hand, the required minimum number of HNs to be placed is determined by the diameter of the structures, the number of substructures, and the given communication range of the HNs. On the other hand, the minimum and reliable communication range can be set by the number of available HNs', the diameter (width also can be considered) of each substructure, and the total number of substructures. The communication range should be adjustable. This implies that, normally, the required number of HNs' placement is proportional to the number of substructures.

However, h_n (the number of required HNs) can also vary due to the wireless link quality ($Q_{u,v}$). This is because there should be more than one HN placed in a substructure when placement of only one HN shows low $Q_{u,v}$. We then look for the locations with high location quality indicators for the HNs' placement, and find high link qualities ($>Q'_{u,v}$) within distance $d_{u,v}$ from one sensor location to another.

The implementation procedure in Algorithm 1 for determining the optimal locations involves working with all measurement points/locations of F by generating a data matrix (step 1). In step 2, E_j, which is associated with the data matrix, is computed and all of the locations are sorted according to E_j. In step 3, first a location with high location-quality indicator is required to be set. The location can be searched from the location of the sink. The algorithm finds each location from M with E_j and $Q_{u,v}$. In this case, if there are any neighboring nodes (HNs) already placed, link qualities of a sensor location are compared with the locations of the nodes. Otherwise, any locations with high contribution to E_j are chosen and link qualities are compared.

When each such location (satisfying both high location-quality indicator and high link quality) is found, an HN is placed there. This placement ensures that each placed

ALGORITHM 2: Optimal Location Finding for LNs

Input: Given a set $L(\subset P)$ of n_l LNs, $M - n_h$ candidate locations.
Output: Location assignment for the n_l LNs.
Step 1: get Φ^m with dimensions $n_l \times (M - n_h)$;
    get data matrix X;
Step 2: **for** $i = M - n_h$ *to* $n_l + 1$ **do**
        compute E_j;
        sort remaining locations $M - n_h$, according to E_j;
    end
Step 3: **for** $j = 1$ *to* n_l **do**
        find respective E_j for each jth location after removing the respective row and column;
        cache the update after removing;
        place j at s_j;
    end
Step 4: **if** $j = n_l$ **then**
        remove the location from $M - n_h$ that is with minimum location quality to E_j;
        set the next location s_j according to E_j;
          if $i = n_l$ **then**
            break;
          else
            go to Step 2;
          end
    else
        find location s_j for the next jth sensor;
        go to Step 3;
    end
  return placement of n_l;

HN has the high link quality to at least one of its neighboring HNs that should be within distance $d_{u,v}$. Finding locations for HNs out of M has a very low complexity $O(n_h M)$, because there are no other nodes placed with which to compare. The high complexity of an HN placement is $O({n_h}^M d)$. Here, n_h is the number of HNs and d is the maximum distance from an HN to another HN, or the sink. Practically, n_h is from few to many in a SHM system. As described earlier, the required value of n_h can be defined by an SHM end-user.

4.2. The Second Phase: LNs Placement

In this phase, we place all of the given LNs. Algorithm 2 is given for the LNs' placement. The initial implementation procedure is similar to Algorithm 1. In Algorithm 1, we compare minimum link quality in distance $d_{u,v}$ during the placement of HNs. On the one hand, when considering the high link quality, we may miss some of the locations with high location-quality indicators. On the other hand, there may be more than one sensor location within $d_{u,v}$ that may have both high location-quality indicators and link qualities. Such locations may be missed due to the limited number of HN placements in a substructure. We have to ensure the placement of sensors at these locations.

Thus, in Algorithm 2, we do not estimate $Q_{u,v}$ within $d_{u,v}$ of any two LNs, or an LN and an HN. The LNs are placed by fully finding the locations with high-quality indicators out of $M - n_h$ locations. This ensures that all of the optimal locations can be found for monitoring. A question may arise that sensors placed at some of such locations may not satisfy high link quality, or some sensors placed at some of such locations may be poorly connected. We address this concern in Section 4.3.

In Algorithm 2, step 1 is similar to step 1 in Algorithm 1. In step 3, we compute E_j for each LN. This is done by deleting the rows and the columns of the matrix that correspond

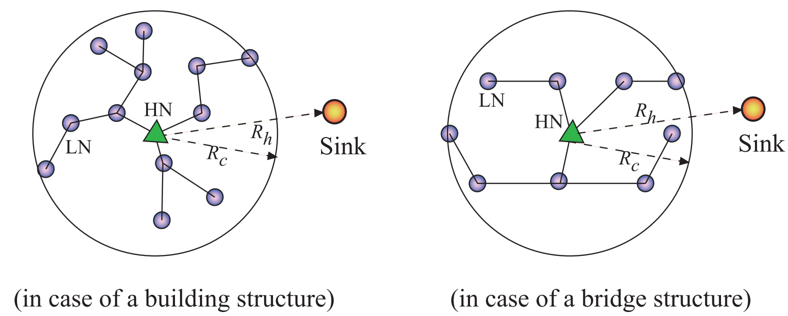


Fig. 2. Illustration of an HN-LN tree (a distributed view in a WSN).

to the sensor measurement with the lowest noise effect (step 4). Locations with low contributions to E_j are removed from the candidate location set. LNs are placed one by one at the remaining locations after removal. Compared to a thorough search, this solution reduces the complexity from $O(n_l^{M-n_h})$ to $O(\sum_{i=1}^{n_l}(i \cdot (M-n_h) + i \cdot n_n))$, where n_n is the maximum number of neighboring sensor locations, including some HNs and some LNs that are already placed. We give a bound on the complexity by $O(n_l^3 M)$.

4.2.1. Grouping and Connectivity Maintenance. There are more challenging issues when forming a WSN by placing sensors via EIM, for instance, communication, connectivity, and fault tolerance. We address these issues to provide a high-level network performance for SHM.

The idea is to make groups of closely placed sensors and to connect each group of sensors by an HN instead of connecting each sensor to all of the others in a group, as shown in Figure 2. LNs in a group that are within R_c of an HN, and one or more other HNs within R_h, can be covered by (connected to) the same HN. The covered groups can then be connected to their closest neighbors to provide stable connectivity of the network. In addition, in order to find possible vulnerable points (the locations at which the placed sensors are poorly connected) in the network, this sensor grouping is crucial. Figure 2 illustrates the grouping of HNs and LNs, and covers each group by at least an HN. We present Algorithm 3 for grouping the sensors and maintaining the connectivity.

Algorithm 3 includes four steps. In step 1, LNs find their neighboring nodes with R_c and connect them. Step 1 shows that every LN $(u, v, w, \ldots)$ finds its required closest neighbors and connects them. r_{nb} is a variable that denotes the number of required neighboring nodes to which a node should connect. Suppose that $r_{nb} = 3$ in a WSN system. This implies that an LN is restricted to connect more than three neighboring LNs. However, $r_{nb} = 3$ can be more or less, depending on the structural environments where the network is being deployed.

Step 2 further confirms the connectivity between any two LNs. Step 3 shows the connectivity between LNs and HNs. There is also a restriction that an LN must connect at least one HN. This is given for the purpose of fault tolerance during monitoring.

Step 4 is very important, as it establishes a connectivity tree, called an "HN-LN tree." The idea is to connect all of the LNs to HNs. More specifically, the tree is constructed from a set of LNs to a root, namely, an HN (see Figure 2), which resides in an HN's R_c. The LNs that belong to an HN collect the structural data and send it to the HN. In other words, an HN-LN tree is constructed by the flows, where a *flow* is a connection between an LN and an HN. Each HN maintains a separate HN-LN tree and records all of its LNs' information in the local memory. Intuitively, such a tree can be considered as a *subnetwork*, that is, the portion of the structure that is monitored by a subnetwork can be viewed as a *substructure*.

ALGORITHM 3: Grouping and Finding HN-LN Connectivity

```
Input: Given a set H of n_h placed HNs and a set L of n_l placed LNs.
Output: Connected network and HN-LN trees.
Step 1: n_n = 0;    //the number of neighboring nodes
        for each LN u in L do
            for each u, v, or w in L do
                if (d_{u,v} ≤ R_c) and (d_{u,v} ≤ d_{u,w}) then
                    u connects v as a neighboring node;
                else
                    u connects w as a neighboring node;
                end
                n_n ← count the number of neighboring nodes;
            end
            if (n_n > r_nb) and (d_{u,v} > d_{u,w})
            then
                replace the furthest neighbor with a closest one;
            end
        end
Step 2: for each v is a neighbor of u do
            v connects u as a neighboring node;
        end
Step 3: n_h ← 0;    // the number of HNs
        for each LN u of L do:
            if n_l ≤ r_nb then
                connect any HN h is within R_c;
                if u is not connected to h
                then
                    u sets h as its HN;
                end
            end
            n_h = n_h + 1;    //the next HN
        end
Step 4: for each u is an v's neighbor do
            add u to the HN-LN tree;
            for each neighbor v is an u's neighbor do
                add v to the HN-LN tree;
            end
        end
    return connected network and the HN-LN trees;
```

In step 4, during grouping of the LNs in the HN-LN tree, the LNs that are farthest from an HN (i.e., in the boundary) may be overlapped and covered by (connected to) more than one HN. The covered trees can then be connected with each other to provide full connectivity of the WSN. The complexity of the grouping algorithm is $O((n_l + n_h)^2)$, where n_l and n_h are the number of LNs and HNs, respectively. This is because each LN is checked for grouping with all other LNs and HNs.

4.3. The Third Phase: RNs Placement

The WSN formed by the LNs and HNs placement in the first and second phases may have connectivity during the network runtime, as long as all HNs are functional. Each optimal location is monitored as long as all LNs are functional. Nevertheless, we still have several concerns to address in a WSN placement for SHM.

—Through Algorithm 1 in the first phase of placement, it is possible that once HNs are placed with high link qualities, the deployed HNs may connect some of their neighboring nodes with the lowest communication costs. However, when using the distance-based model for link qualities, we need to pay attention two exceptions to: (i) We may still miss some of the locations with high-quality indicators. (ii) It is impossible to have reliable links at some locations with high-quality indicators in different structural environments, in some instances. For example, there are physical structural constraints (see constraint C5 in Section 3.4) and there is physical interference. We need to simultaneously ensure that both high-quality locations and low communication costs are kept.

—Through Algorithm 2 in the second phase of placement, there may be a question as to whether sensors placed at some locations with high-quality indicators have poor link qualities (using the distance-based model) or sensors placed at those locations are poorly connected (or not connected at all). We have to improve the link quality to the sensors placed at these locations.

—There are some events that may occur during the network runtime: (i) If the link quality from a node u to a node v is poor, many times v's location may not be monitored. (ii) If an LN fails, the locations may not be monitored; if an HN occasionally fails, a region of the WSN may fail. For example, in Figure 2, a failure of one LN or HN may cause a region of a subnetwork (or an entire subnetwork) to become disconnected from the WSN.

A general approach to improve the preceding concerns in a WSN is to place some Redundant Sensors (RNs), as required, at some locations so that the following benefits can be achieved: (i) The optimal location can be guaranteed to be monitored long term. (ii) High-quality links can be ensured over the network. (iii) Some LNs that are located at the optimal locations may have one or more sensors to communicate with the HNs. (iv) Reduction of the longest-distance communication and its load is obtained.

We can define the level of redundancy as the number of mutually exclusive paths (k) from a sensor to the sink. The level of redundancy can be estimated as the minimum level of redundancy. We call the RN placement solution as k-redundant and its level of redundancy is k. This depends on the following criteria: (i) any sensor has at least k exclusive (alternative) routes to its sink; (ii) a sensor is located in the boundary and connected to at least two HNs.

We put the priority of finding locations for RNs in accordance with the aforesaid criteria:

—The first priority is to place an RN between an LN u and another LN v, between an LN u and an HN v, or between an HN u and another HN v, that is, to mitigate the low link quality between any two nodes u and v by placing RNs at relative locations if the obtained link quality is lower than the acceptable reliable link quality. In other words, $Q_{u,v} \leq Q'_{u,v}$, see line 6 in Algorithm 4.

—The second priority is to place an RN (as an intermediate node) between any two nodes u and v at a relative location if the distance between the two nodes is longer than expected. The target is to mitigate the communication load by minimizing the maximum distance d. That is, $d_{u,v} \geq R_c/2$, see line 11 in Algorithm 4. In fact, there may not be found many such two node locations. In some instances in the first priority, a low link quality between any two nodes (where $d_{u,v}$ between them is longer than expected) can also be improved. Thus, when the high link quality between the nodes is ensured, the long distance between them is also minimized by the first priority.

—The third priority is to place an RN at a junction LNs' location. In multihop communication, such a node may fail due to being a data forwarder for its descendant nodes (lines 15 through 18 in Algorithm 4).

ALGORITHM 4: Priority-based Redundant Sensor Placement (PRSP)

Input: A set R of n_r RNs, $M - (n_h + n_l)$ candidate locations.
Output: Placement of the n_r RNs into the WSN with HNs and LNs.
begin
1: $r \leftarrow$ the number of RNs that are already placed; //initially, $r = 0$
2: **if** $r \leq n_r$ // if there is an RN is available
3: **then**
4: **for** *each placed HN* **do** // until all of the given HNs are visited
5: **for** *each LN belonging to an HN* **do** // until all of the LNs are visited
6: **if** $Q_{u,v} \leq Q'_{u,v}$ **then**
7: **Call X**
8: **if** (*there is an s_r available out of M_r*) **then**
9: place an RN at s_r;
10: **end**
11: **else if** $d_{u,v} \geq R_c/2$ **then**
12: **Call X**
13: **if** (*there is an s_r available out of M_r*) **then**
14: place an RN at s_r;
15: **else if** (s_u *is of an LN location* and s_v *is with only one sensor*) **then**
16: place an RN at s_v;
17: **else if** (s_v *is of an LN location* and s_u *is with only one sensor*) **then**
18: place an RN at s_u;
19: **else**
20: place an RN at s_u; // boundary or intermediate node
21: place an RN at s_v; // boundary or intermediate node
22: **end**
23: **else**
24: the next link; //another link from u to v
25: **end**
26: **end**
27: **end**
28: increase r; // r increases as an RN is placed
29: **end**
30: **return** placement of $n_r (r \leq n_r)$;
end

X:
find location coordinates s_u and s_v of node u and node v;
$M_r \leftarrow 1$ to $M - (n_h + n_l)$ //remaining locations;
sort remaining locations M_r according to E_j;
find a relative location s_r between s_u and s_v;

—The fourth priority is to place RNs at the boundary LNs' locations, which may be connected to at least two HNs, thus having higher transmission cost (see lines 19 and 20 in Algorithm 4).

We assume that a location may be or may not be on the straight line from an LN to its HN for each RN, as shown in Figure 2. Intuitively, such a placement offers exclusive routes from each LN to an HN. We propose Algorithm 4, called *priority-based redundant sensor placement* (PRSP), based on the aforesaid priorities, to meet our system objectives.

The PRSP is a distributed sensor placement algorithm: the RN placement problem in SHM involves finding a location assignment for the given set R of n_r redundant sensors out of $M - (n_l + n_h)$ possible locations, subject to the connectivity and data delivery constraints. At the beginning of the PRSP there are no RNs placed, namely, $r = 0$. The algorithm runs until the number of given RNs (i.e., n_r) is placed into the

WSN that is with the HNs and LNs. After the placement of one or two RNs, the PRSP checks whether we have RNs available or not, where $n_r < n_l$ and $n_r < M - (n_l + n_h)$; we omit some steps in the algorithm.

The PRSP executes via an HN to another HN, one by one until all of the HNs are visited (line 4 and line 5). In line 6, the obtained link quality is compared between any two nodes u and v, and an RN is placed if the obtained link quality is lower than the acceptable reliable link quality. An LN connected to which LN, or belonging to which HN, can be determined by comparing $d_{u,v}$ (line 11).

It is obvious that $n_r < M_r$, where n_r is the number of RNs in R and M_r is the number of remaining locations. If there is a near-optimal relative location, namely a location having $E_j > 0.4$, a sensor is placed at the location. If there is no location available, or $E_j < 0.4$, but there are more RNs available to be placed, we place each RN at the same location of an LN. An RN is placed at an empty location or at a location with having only one sensor placed, that is, we make sure that at most two sensors can be placed at any location.

The complexity of the PRSP algorithm is $O((n_h + n_l) \cdot d)$, where n_h and n_l are the number of HNs and LNs, and $d(> R_c/2)$ is the maximum distance between any two sensors u and v. This expression is taken by the following observation: for a group of LNs ($O(n_l)$), the two coordinates of an RN are computed and the RN is placed, until the RN is within R_c of an HN in $O(d)$ iterations. In a similar way, we can show that the number of RNs placed is with $O(n_l d)$. The total complexity depends on how the RNs are placed, and n_r. It is difficult to give an analytical assessment of the number of RNs that can be placed by the PRSP, because the number of RN placements depend on how the LNs and HNs are placed, n_l, and n_h.

5. ANALYSIS OF CYBER-PHYSICAL ASPECTS AND FAULT TOLERANCE

This section analyzes and addresses some aspects related to the cyber-physical system (CPS). These aspects include fault tolerance through communication and connectivity optimization, structural mode shape normalization, etc.

5.1. Fault Tolerance

5.1.1. Communication and Connectivity Optimization. We have already placed all of the given sensor nodes in systematic ways for SHM. Although the strategy of direct transmission may be considered reliable in generic WSNs, we do not use direct transmission from an LN to the sink in this system because of cost inefficiency. In wired-based centralized SHM, all kinds of analysis are performed offline at the sink. It is difficult to do so in WSNs, as they are prone to fault or failure.

In SHM application, each sensor is required to collect data at a high sampling frequency such as X00 times/s, X=1,2, ..., while it is X times/s in generic WSN applications. In practice, if a sensor transmits such a large amount of data (i.e., information about all mode shapes) to the HNs or the sink, its T will reduce quickly because of energy depletion for communication. In this cyber-physical system, LNs are configured to compute locally using a decision-making algorithm [Bhuiyan et al. 2012a] rather than transmitting data over a large structure. They transmit their refined data (i.e., aggregated mode shape) which is small in amount (e.g., a few to X bytes) compared to raw data (e.g., generally more than X000 bytes).

The system not only suffers sensor faults or failures, but also from routing inconsistency and connectivity problems in WSNs, where wireless communication is less reliable in the structural environment. During the deployment, we provide an extent of communication efficiency through the three-phase sensor placement (finding possible locations with reliable communication links and requiring a minimum number of unnecessary retransmissions). Also, we provide a level of fault tolerance in the WSN through RNs placement in the third phase. However, we are still required to provide fault tolerance during network runtime, allowing for some situations such as the following.

—If the number of RNs is limited, we cannot provide RNs at all of the optimal (e.g., $E_j \geq 0.7$) or near-optimal (e.g., $E_j \geq 0.4$) locations. These locations are important for SHM, or at required locations (e.g., junction or articulation point) that are important for WSNs. Such locations may have only a single sensor.
—In some events, for example, a location is damaged or there is an event of earthquake, the LNs may be requested by an HN to work for a prolonged period of time. As a result, they may fail during operation.
—Communication links with a wireless sensor are not reliable. We think that links can be highly dynamic and bursty in structural monitoring environments. The drop of data packets can impair the quality of the monitoring. A packet loss means the mode shape information (which may contain interesting data about damage) is lost. It is impossible to distinguish damage and its approximate location if some of the transmitted packets are missing. In the situation of sending an "alert," an SHM system poses additional reliability requirements. Such problems exist in generic WSN applications, which often consider a flat sensing field with moderate dimensions. However, the deployments reach an extreme by the pathological extension along the horizontal or vertical dimension in a large structure monitoring.

We provide network maintenance (repairing) during the network runtime to mitigate such situations and ensure monitoring the optimal locations. We argue that the real-time and local maintenance better suits WSNs since they are asynchronous and reactive in nature.

5.1.2. HN-LN Tree Maintenance. If an LN fails, we assume that an HN is aware of the failure, since each HN has the list of LNs through the HN-LN tree and also neighboring HNs. Each HN strives to keep the HN-LN tree information up-to-date. An HN can locally maintain the connectivity through the HN-LN tree. When the LNs report on the refined data to its HN, the reports may be similar or different. However, the health state is guaranteed to be known and monitored by at least one HN, even if $k-1$ LNs or an HN fails. When data packets sent by LNs to HNs are lost in the wireless link, alternative links attained by connectivity maintenance (e.g., k-connectivity) can also be provided to tolerate data-packet loss. Missing data packets in multiple consecutive times from an LN are used to detect the failure of the LN.

Definition 5.1 (*k-Vertex HN*). The WSN G is k-vertex HN-connected if there are k pairwise vertex disjoint paths from any LN u to at least one HN. Equivalently, the WSN is k-vertex HN-connected if the removal of any $k-1$ LNs (and all of the related links) does not partition the network. That is, there will be a path from every LN u to an HN.

Depending on the HNs' locations in the network topology obtained by our deployment, no further node placement is needed. Let E be the set of all edges of G. We estimate a *distance-weight function* on the edges [Cardei et al. 2008] that guarantees that two edges with different end-nodes have different distances. Under constraints C1 to C3 (described in Section 3.2) and Definition 5.1, when a sensor periodically reports its decision data, the data delivery must be fault tolerant to the failure of up to $k-1$ nodes. We use Algorithm 5 for network maintenance (or optimization on communication reliability and connectivity) in case of sensor faults and routing inconsistencies.

Algorithm 5 has two main steps or two subalgorithms. Step 1 repairs the network in case of sensor faults and link faults. If any routing inconsistency is found, it is validated. Step 2 maintains the connectivity in the network. In a WSN, an HN runs this

ALGORITHM 5: Optimization of Connectivity and Routing Inconsistency on HN-LN trees

```
Input: Connection on the HN-LN tree.
Output: k-Connected HN-LN tree.
Step 1: Sort all of the edges in a tree in order of distance to the HN;
        for each sensor node f in HN-LN tree do    // f denotes a faulty/failed node
            mark f as a failed node;
            find the nearest LNs of f, hit the links by PING protocol;
            check LNs and RNs list for finding an RN at or near the f's location;
            if there is an RN available then
                activate the RN and connect it to the HN-LN tree;
                go to Step 2;
            else
                find an alternative link and route maintenance;
                request the LN located near the f to adjust R_c by choosing τ to cover f's location;
                go to Step 2;
            end
        end
Step 2: k-connectivity maintenance;    // e.g., k ≥ 2
        for each edge (u, v) in the sorted order do
            duplicate all its edges from a node u to any node v toward the HN in HN-LN tree
            if u is k-vertex connected to the HN then
            then
                set each edge as an alternative link;
            end
        end
    return k-vertex connected HN-LN tree;
```

algorithm in a distributed manner, whenever it is aware of three situations: (i) an LN is failure; (ii) a connection between two LNs toward an HN or an LN and HN is broken; (iii) there is data-packet loss consecutively due to link failure or other reasons.

An HN needs to handle the situations, because structural damage may occur in a specific subnetwork that should be covered fully by the HN, and partially by one or more HNs. This implies that the whole WSN does not need network maintenance. The sink node may be informed by the HN about the maintenance, but it does not need to intervene with the maintenance. Thus, $C.cost$ (communication cost) as well as T will not be affected.

In step 1, all of the edges are sorted according to the distance weight and masks the faulty or failed node. An HN has the information of the failed LN and the closest neighbors of the failed LN. It communicates to these neighboring LNs and checks if there are any available RNs in the sleeping mode. If there is an RN, the task is to provide a scheduling technique for the RN's activation with a bounded latency. When an LN fails, a corresponding RN that is placed at the same location should be activated in a time bound, and should be connected to an HN. It ensures that vicinity of the failed LNs will be monitored. T can be longer as well.

If there is no RN available, the HN requests those neighboring LNs to adjust both R_s and R_c up to cover the location of the failed LNs. There may be routing inconsistency in the tree due to the link faults and sensor faults, or associated reasons. We provide route maintenance to overcome such a routing inconsistency. We apply the technique of datapath validation to account for the inconsistency [Gnawali et al. 2006].

The technique enables routing layers to remain reliable in highly dynamic topologies on many different link layers. In the technique, each data packet from a node contains the distance to its neighboring nodes. Routing inconsistencies are perceived when an alternative route is needed, even when the control traffic rate is very low. The technique is suitable for validating the routing path problem in the HN–LN tree in a subnetwork as it is localized.

Step 2 is about optimizing the k-connectivity on the HN–LN tree to a desired connectivity, for example, $k = 3$. On the one hand, when an LN fails, at least two links may be broken that correspond to the failed LN. On the other hand, when one link is broken, the HN can choose one of two alternative links that entails the minimum cost. In step 2, similar to the connectivity algorithm in Cardei et al. [2008], at first, an HN duplicates all the edges on the HN–LN tree. It examines all the edges in decreasing order and removes an edge (u, v) if, after its removal, node u remains k-connected to the root HN. Then, the algorithm computes its R_c for any LN such that the LN can directly communicate with any other LN that is joined by an edge in the HN-LN tree. The maintenance process can continue until reaching k-connectivity.

By using the WSN flow technique, a query on whether or not two vertices are k-connected in the HN-LN tree can be answered in $O(E + V)$ time for any fixed k [Cardei et al. 2008]. Therefore, the complexity of Algorithm 5 is $O(e(e + n)) = O((e))^2$, where e and n are the number of edges and nodes in the HN-LN tree, respectively.

5.2. Communication Cost (*C.cost*)

In the WSN, we calculate the number of packet transmissions in two cases: (i) A refined decision packet from an LN travels through an intermediate LN or directly to an HN. The HN acknowledges to the LN that the decision packet is received or informs of a corrupted packet: (ii) the amount of packets that exchange between nodes for the network maintenance in achieving required fault tolerance, routing validation, and connectivity improvement.

If a packet travels across three links until it reaches the common ancestor HN or the sink, there will be a cost of three packets per unit time. Note that these three links are comprised of the only path in between the two indicated nodes. We thus define $C.cost$ as the total communication cost in the WSN. Let δ be the HN-LN tree hierarchy as the sum of the individual communications between all pairs of nodes adjacent in G.

$$C.cost(G, \delta) = \sum_{(u,v)\in E_G} \omega(u, v) \cdot pathCost_\delta(u, v) \tag{4}$$

Since neighboring nodes in the same HN–LN tree may be physically distant, we define the costs of the tree links used in the $PathCost_\delta$ computation to be $d_{u,v}$. Thus, the $C.cost$ reflects the required radio energy consumption. $C.cost$ includes the cost of the total number of retransmitted packets and acknowledgments in instances of dropped packets. Thus, $C.cost$ increases as the packet drop rate increases, that ts, the higher the packet drop rate in a system, the higher the $C.cost$ in the system. To have a understanding of the packet drop rate, we provide an example in the following.

We investigate the impact of network congestion on data dissemination in different experimental WSNs. Our interest is on the packet drop rate in the SHM environment. In TPSP, we conduct an experiment on an outdoor structure with 22 Imote2 sensor nodes. The sensors are placed on different locations of the structure. The experiment is performed over a 7-hour period. Each node is with a 3-axes accelerometer, sampling at 560 Hz with a 140 cutoff frequency (see Section 7 for more details).

SPEM finds 11 sensor locations, selects eight locations among them on the eight floors, and places 16 sensors. Two sensors are placed on each floor. SPEM collects acceleration data at 560 Hz for 24 hours [Li et al. 2010]. It integrates topology control and data routing techniques with the SHM framework, where all of the sensor nodes must be connected and the data routing in the WSN has a many-to-one pattern.

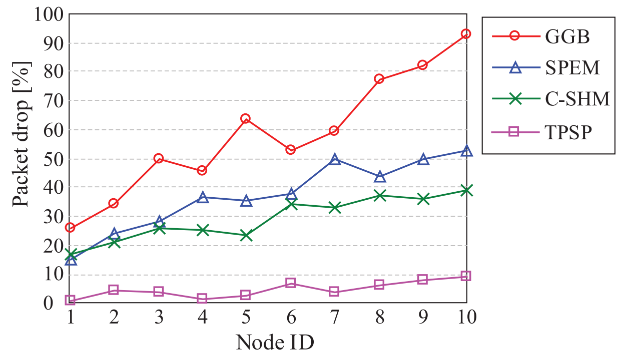


Fig. 3. The maximum amount of data packets dropped by the WSN per packet transmitted to the sink as a function of the transmission rate.

Another approach, C-SHM, conducts experiments on a small-scale structure with reduced and adjustable transmission power levels [Liu et al. 2011b]. The structure has 10 floors; at each floor, an Imote2 is deployed to sample the structure's horizontal accelerations in a synchronized manner at a frequency of 512 Hz. Each sensor connects its one-hop neighboring nodes within the transmission range, and transmits the data hop-by-hop. A WSN with 64 nodes is deployed on the 4200ft-long main span and tower of the Golden Gate Bridge (GGB) [Kim et al. 2007]. The nodes collect ambient vibrations data synchronously at a 1000 Hz rate. The sampled data is collected reliably over a 64-hop network.

We analyze the number of packets sent by each node and determine the *packet drop rate* for the first 10 nodes with regard to the sink location, illustrating the results in Figure 3. Note that C-SHM is with the network of 10 nodes. We calculate the maximum packet drop rate over each link for each node. It is essential to remark that this rate may vary due to the specific environments and the interference in the environment, distance from the sink, communications techniques, etc.

In GGB, the nodes closer to the sink experience a packet drop between 27% and 50%, while the last node experiences a packet drop rate over 90%. The maximum packet drop is about 53% at the last node in SPEM, 37% in C-SHM, and 9% in TPSP. The average packet drop rate found in TPSP is 4.1%. Figure 3 shows that, as the packet transmission rate increases beyond a certain network capacity threshold, congestion occurs more frequently and the number of packets dropped per received data packet at the sink increases remarkably. For example, 7.5 packets are dropped across the network for every data packet received at the sink in GGB, while it is 0.3 packets in TPSP. This may be one of the reasons that GGB requires 9 hours to collect a single round of monitoring data from the network of 64 sensors. This system's large latency may arise from the fact that the SHM method is centralized/global and is designed separately from the WSN system. In TPSP, the experiment is repeated the next day with all of the nodes placed at the same locations. We found similar results.

5.3. Structural Mode Shape Analysis under the WSN

5.3.1. Reference Mode Shapes and Overlapping Mode Shapes. In this section, we discuss what kind of information sent by the nodes in the WSN toward the sink, how this information is produced, as well as what are the associated difficulties in the collection. We use a distributed damage detection algorithm suitable for WSNs proposed recently by Bhuiyan et al. [2012a] to analyze the performance of mode shape under sensor faults. However, existing distributed damage detection algorithms can also be utilized in this approach [DeVore 2013; Hackmann et al. 2013, 2012; Rice and Spencer 2009; Ni et al. 2008; Xing and Mita 2012].

In the algorithm, the signal processing is implemented in each sensor node. Each node continuously analyzes the accelerometer samples $\{a_x\}$, $\{a_y\}$, and $\{a_z\}$ captured in each consecutive time window through two procedures: (i) by counting the maximum absolute value of the vibration velocity and its corresponding timestamp; and (ii) the dominating frequency of the vibration velocity spectrum for each coordinate $i = x$, y, and z. The parameters are then compared with a *reference value* (or a vibration-threshold).

Based on the comparison, the node needs to deliver a decision (either "1" for an unusual condition or "0" for a normal condition). Besides, the nodes are enabled to compute a local aggregated mode shape based on its collected samples and compare it with a *reference mode shape* (which will be described later). Thus, the two kinds of information are normally delivered to the HNs for a possible damage detection in a substructure. The HNs also deliver the final aggregated information based on the decisions transmitted by all of the nodes in the substructure.

If a node makes a decision "1" and the aggregated mode shape largely differs from the reference one, the nodes store the parameter values and all acceleration samples captured in the time window. The amount of data transmitted over the network (that is placed by the proposed placement algorithms) is small. In an exception, the node can be issued an "alert" about possible damage by an HN. In such an exception, the node should deliver all of the values and samples through the HN upon request by the sink. In such an exception, we use a Huffman compression method to reduce the amount of data delivered [Huffman 1952]. In the exception, we check that the amount of data transmission by nodes placed in a substructure does not exceed the given channel capacity threshold (although we consider an adjustable channel capacity in our work).

An HN can examine whether or not there are significant changes in the mode shape. If there is a "damage" state, it makes a request to the LNs in the subnetwork to continue taking measurements and transmitting the mode shape. Meanwhile, it transmits an *alert* to the neighboring HNs and the sink. Each HN can execute the command as requested by the sink. There are three reasons for decision sharing with the neighboring HNs, described as follows.

—There is a chance that an LN can be connected to two or more HNs. Thus, two or more HNs may (or may not) have the same decisions (i.e., *alert* when there is a change in the mode shape of a substructure). Thus, each can buffer the decisions received from the neighboring HNs and compare them.
—If a damaged location is in the boundary area, the LNs overlapped by the two HNs can get the same decision from the LNs. In this way, the damaged area can easily be localized and the sink is able to analyze exactly the location of the damage.
—In case of an HN failure, the health state can still be transferred to the sink through the neighboring HN. Some LNs that belong to different HNs can be overlapped (as shown in Figure 4) in this cyber-physical system.

Since some sensor nodes may overlap with each other, the mode shape can also be overlapped. Thus, we should tackle this. We need to normalize the overlapped mode shape and optimize the communication between overlapping LNs and HNs.

5.3.2. Mode Shape Normalization. We not only estimate information about locations and mode shape based on EIM, but also normalize it for different subnetworks in the WSN. As described earlier, the covering area of an HN is the area of a substructure. The *true mode shapes* should be identified in the majority substructures, while the *noise modes* will randomly appear in the substructures. We consider that each sensor is given a *reference mode shape* of the vicinity of its location that it can compare with the

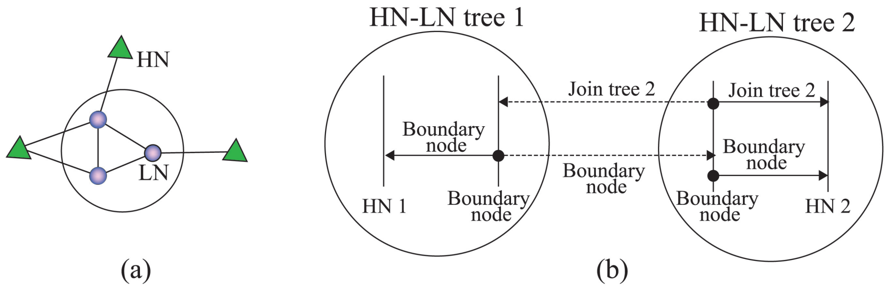


Fig. 4. (a) Boundary nodes; (b) communication between boundary nodes and HNs to optimize connectivity and inter-HN–LN-tree connectivity, and normalize mode shape.

identified mode shape during monitoring. A reference mode shape is the aggregated mode shape of the structure when the structure is in the normal condition (i.e., without damage).

In this work, the creation of the reference mode shape at the sink is not based on the direct raw data transmission (from the node toward the sink), but based on aggregated local mode shapes achieved through the network. Since each node is required to provide its location-quality indicator during the placement by using the mathematical model shown on the Appendix C, we also enable the node to compute and set the reference mode shape through the same model (without requiring other models or further computation) at the initial system setup. Each HN collects all of its LNs' mode shapes during the initial system runs. Then, the mode shape information of a substructure can be found by the mode shape aggregated by an HN. Based on all the mode shapes collected from all of the HNs, the sink can have the reference mode shape of the whole structure. Thus, when a specific mode shape is identified, without noise, it is considered as a true mode shape. Without having a true mode shape, it is difficult to disambiguate the actual health state from a faulty sensor's decision.

Creating localized mode shapes through the HNs can benefit a WSN-based SHM system in three ways.

—It can reduce the amount of time for creating a reference mode shape. If the reference mode shape is computed at the sink, based on the direct data transmission, it may take more time than a normal monitoring time window to transfer all of the samples at the sink. This is mainly due to wireless traffic and channel capacity constraints. We think that it can interrupt the initial monitoring operations and the overall quality of monitoring.
—It can enhance the reliability on the damage detection. If some of the data packets cannot be recovered at the sink, it is not possible to create a true reference mode shape. The monitoring operations may be meaningless due to incorrect information in the reference mode shape.
—It can enhance the chance of damage detection. Vibration accelerations can vary from one location to another, from one substructure to another, and from one environment to another. If a global reference mode shape is captured at the sink, it may reduce the chance of capturing the variation at some specific locations or substructures of the structure. Creating the reference mode shape by each node locally, and by each subnetwork placed in a substructure, may help overcome this.

In other words, SHM requires correlated data because a damaged area may spread over many sensors' measurement areas. Generally, in a centralized SHM system, tackling an event of a sensor fault or failure, or data-packet loss, is costly. Such events further enhance the network data traffic. In this cyber-physical system, if a sensor fails and there is no RN available at the failed sensor location, the damage information can still be collected by overlapping sensors. In this case, we need to improve both the coverage and connectivity to tolerate the failure and to collect the mode shape of the faulty sensor location. Since we consider a distributed solution, we address the mode shape normalization allowing for the overlapping area in the WSN to assess the actual health state accurately. A more detailed explanation of the mode shape normalization can be found in Appendix D.

6. PERFORMANCE EVALUATION

6.1. Methodology

We first evaluate our proposed algorithms in TPSP through simulations. While the algorithms are general, we adopt the parameters of the sensors in our simulations similar to the Imote2 [Crossbow Technology 2007]. We use real datasets collected by the SHM system on the GNTVT, a high-rise infrastructure monitoring project of Hong Kong PolyU (see Footnote 1 in Section 1) [Ni et al. 2009]. Our objective is to observe the performance in different metrics: placement quality (E_j), network lifetime (T), communications cost (*C.cost*), fault tolerance, and mode shape (Φ) identification.

We consider ($N =$) 110 sensors (including 12 HNs, 80 LNs, and 18 RNs) for the simulations. The algorithms in TPSP are mainly designed to deploy a heterogeneous WSN (with an intention to have a high-quality and long-term structural monitoring). However, we also evaluate the algorithms in a homogeneous WSN placement and compare the network performance results with the results achieved under the heterogeneous WSN. In the homogeneous WSN, all of the sensor nodes have the same capabilities. We define the sensor capability in terms of the amount of energy and the transmission range. In the heterogeneous WSN, the HNs are given double the amount of energy and communication range of LNs.

In the simulations, all geometric properties in the datasets are exactly adopted without further approximation. At the initial setup, a mode shape computed by each node is stored locally as a reference mode shape. Since a node is required to provide its location-quality indicator by using the mathematical model shown in Appendix C, we enable the node to compute and set the reference mode shape through the same model. Then, we implement all of the algorithms, including EIM, using the MATLAB toolbox. GNTVT has a total height of 610m, including a 450m main tower. We set the simulation environment to 450m $\times$ 50m.

For the simulation settings, we use the raw FEM model to generate Φ and MATLAB software module (SPEM) of the GNTVT system [Ni et al. 2009; Li et al. 2010]. We modify it as needed in TPSP, since SPEM restricts the module for the N locations. In TPSP, the computational procedure begins with the location for the HNs.

In the WSN, we set $k = 3$ in simulations. R_c is set to a maximum 30m and R_h is set up to 80m. We follow discrete power-level settings of the Imote2 [Crossbow Technology 2007]. Initially, $R_s = 20m$; it is increased to 30m, as there is a need of sensing redundancy, especially in the case of sensor fault/failure. When estimating total energy consumption, we consider all of the sources of energy consumption, including sensing, communication, and node idleness. The estimation also includes the additional energy consumption for network maintenance (e.g., fault tolerance), retransmission due to packet loss, as well as clock synchronization. To observe the additional energy consumption in different approaches, we calculate $C.cost$, which is normalized to 100.

We inject random faults into the WSN to investigate the extent to which Φ is recovered and T is prolonged, and $C.cost$ for recovery from the faults. We estimate *failure rate*, which denotes the fault injection randomly, estimated by the percentage of the number of faulty sensors to the total number of sensors given. If one wishes, *failure* *rate* can also be given by switching off, sensor debonding, minimizing battery power, etc.

We use an improved version of the SnoozeAlarm function (described in Appendix B) for SHM that offers wakeup/sleep-cycle functionality to reduce long-term energy consumption [Rice and Spencer 2009]. We enable the sensing unit to continuously collect the vibration acceleration data and make a simplified decision ("1" for "no damage") and aggregated mode shape. If there is an event, the LNs send an interrupt wakeup message (by using the radio-triggering wakeup module, described in Appendix B) toward the HNs and neighboring nodes. If a node receives wakeup messages, it activates its communication channel at once. Otherwise, sensor nodes sleep for a period of time and then wake up for a relatively short period, during which they can communicate with their neighbors and HNs. In the case of giving higher communication capabilities to HNs, they have more communication tasks to perform, and the duty cycle of HNs is 0.5% longer than the LNs. Otherwise, the sleep/wakeup duty cycle for the radios is the same value for both HNs and LNs.

To obtain simplified decisions and estimates of mode shapes for substructures, high-precision time synchronization is required to avoid phase differences in the sensors' collected data. We use a modified FTSP (Flooding Time Synchronization Protocol [Maroti et al. 2004]) to fulfill the needs of time synchronization in a hierarchy (in the HNs and then LNs of the network). Instead of flooding time-sync messages to the nodes directly, the sink node multicasts a time-sync message to select HNs using the relevant semantics. HNs can transmit the time-sync message down the hierarchy of HNs, thus synchronizing only the required subnetwork of the network.

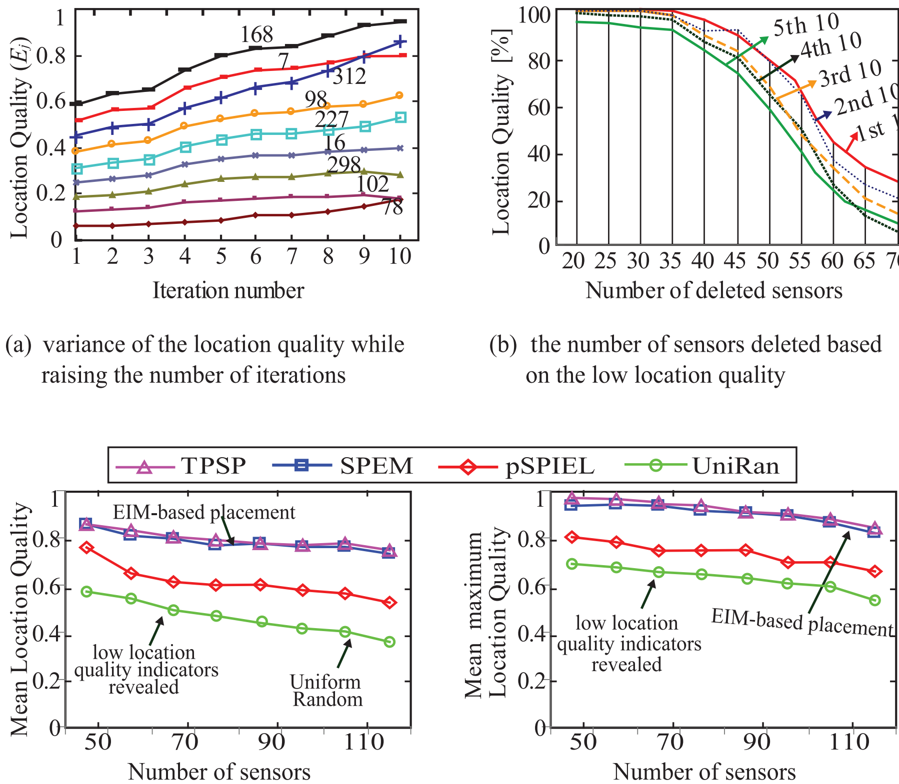
Particularly if there is damage detected in a substructure by nodes in the subnetwork, the subnetwork can be given a priority to be active longer than normal and to collect data for the extended time. The nodes in the subnetwork are synchronized with their HNs, and then the sink that may promise high precision in the event detection in a distributed manner.

For comparison, we implement another three placement approaches.

—*SPEM [Li et al. 2010]* (an approach from SHM domain-specific sensor placement perspective). The SPEM, an EIM-based solution (which is also utilized in TPSP) for sensor placement, exactly the one described earlier.

—*pSPIEL [Krause et al. 2011]* (an approach from both the high-quality location and communication efficiency perspective). As described in Section 2, to the best of our knowledge, pSPIEL is the approach that simultaneously considers sensing location quality and communication efficiency in a realistic scenario. Similar to pSPIEL, the target in TPSP (although it is specially designed for SHM) is to find locations for sensor placement at which sensors may provide both high sensing quality and communication efficiency in practice. In other words, a deployed network by using TPSP should be able to provide high-quality monitoring besides requiring a low communication cost for the monitoring tasks.

—*UniRan* (an approach from both uniform and random placement and hierarchical WSN architecture perspective [Xu et al. 2010]). As described in Section 2, two sets of sensor nodes are placed at the best locations: a set of sensor nodes (like LNs in TPSP) are placed by obeying a uniform distribution to collect sensing data, and a set of relay nodes (called cluster heads, like HNs in TPSP) are placed by using a random distribution to relay the data from the first set to the sink. The approach improves the energy consumption rate problem in the uniform placement method by placing the relay nodes in a random manner, which is finally called the uniform random placement approach ("UniRan" for short). It examines the network performance in both homogeneous and heterogeneous WSN placement cases.



(a) variance of the location quality while raising the number of iterations

(b) the number of sensors deleted based on the low location quality

(c) comparison between the three approaches in terms of location quality

(d) comparison between the three approaches in terms of location quality

Fig. 5. Placement algorithm's performance attained through simulations: (a)–(b) performance of the TPSP algorithms; (c) a comparison of the mean placement quality to the existing approaches; (d) a comparison of the mean of the maximum placement qualities to the existing approaches.

Considering all of the perspectives and the similarities, we choose the approaches for comparison. To make a fair comparison, we consider the same number of sensors (i.e., $N = 110$) for all of the three approaches. In TPSP, we conduct two simulation campaigns: one is for the case of homogenous WSN placement, and one is for the heterogeneous WSN placement. Results analyzed from both cases are compared to the results from all other approaches.

6.2. Simulation Results

We begin by analyzing the simulation results of the placement performance of our TPSP algorithm. In Figure 5(a), a variation of the location-quality indicators for the eight sensor locations as a function of the number of iterations is plotted. Through an iterative procedure in the EIM, locations (such as location 78, location 102) having relatively low-quality indicators are progressively eliminated from the candidate set until the number of remaining locations is considered adequate. The remaining locations (such as location 168, location 7) with high-quality indicators prove more effective in providing sensing quality in SHM.

We observe that sensor location 168 progressively returns the high location-quality indicators as the number of iterations increases. In the simulation results shown in

Figure 5(a), the lowest-quality indicator 0.6 is in the beginning, which is nearly 1.0 at the end. However, in most of the simulations, the sensor location 168 returns location-quality indicators from 0.85 to 1.0. This specifies that this location is one of the best locations for a sensor placement.

The location-quality indicators related to other sensor locations have shown a similar tendency. That is, E_j increases consistently with increased iteration numbers, indicating that their importance is identified through the elimination process associated with the iterative procedure in the EIM. It is important to mention that the sensor locations, which initially show high location indicators but finally show lower location-quality indicators than many other sensor locations, are removed from the sensor location set.

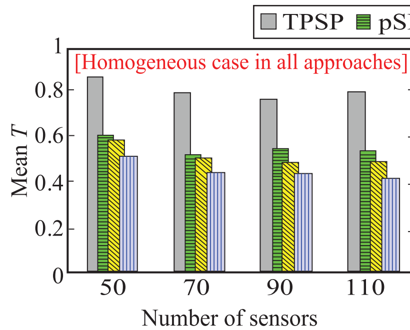
After observing the results in Figure 5(a), we surmise that the total number of sensors deleted are based on the low location quality (the percentage of the location-quality indicator) in Figure 5(b). The corresponding curve depicted in Figure 5(b) goes down when the location quality of a sensor decreases (or as the iteration numbers increase in Figure 5(a), the location-quality indicator decreases). It is evident that, as a candidate location is deleted from the sensor location candidate set, the individual location quality associated with the remaining locations may vary accordingly. We can see that the location quality of the first 10 sensors is higher than the location quality of the second 10 sensors, the location quality of the second 10 sensors is higher than the location quality of the third 10 sensors, and so on. It also indicates how many sensors are eliminated, compared with the location qualities during the placement operation.

Based on the results in Figure 5(b), we show a dynamic performance comparison of the three approaches. We consider the number of sensors (including `HNs`, `LNs`, and `RNs` in TPSP) versus the mean location quality on E_j in Figure 5(c), and the number of sensors versus the mean of the maximum location quality in Figure 5(d). The results are gathered from both simulation campaigns in TPSP. As shown in Figure 5(c), TPSP achieves almost the same location quality as that of SPEM, while SPEM has a better quality than traditional EIM, pSPIEL, and UniRan.

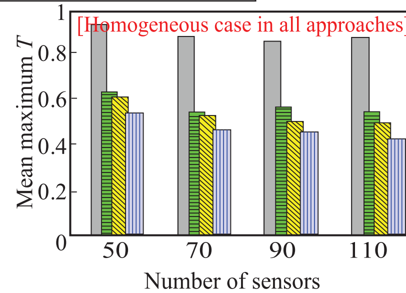
In the simulations, the maximum location quality in a case of 50 sensor nodes provided by UniRan is about 0.713, which is about 0.82 in pSPIEL, 0.93 in SPEM, and 0.96 in TPSP. In the case of 20-sensor cases, location quality in TPSP is close to 1, which is 0.76 in UniRan and 0.84 in pSPIEL. Although pSPIEL provides better location quality than UniRan, both approaches provide poorer results than the TPSP- and the EIM-based approaches. It is evident that the placement approaches (like pSPIEL and UniRan) proposed for generic WSN applications may not be directly applied to SHM applications.

Our next observation deals with T, as shown in Figure 6(a), Figure 6(b), Figure 6(c), and Figure 6(e) in the homogenous WSN placement case of all of the approaches, and in Figure 6(d) and Figure 6(f) in the heterogeneous WSN placement case for TPSP and UniRan. In the first simulation campaign of TPSP, we first estimate the *minimum*, *mean*, and *maximum* lifetime of each sensor in the network, and then estimate the *mean minimum* and *mean maximum* lifetime of the WSN. For comparison, we take into account the mean minimum lifetime (or the worst-case lifetime) achieved in the simulations. We think that the values obtained from the total energy consumption of a sensor node may not always correspond to a maximal system lifetime. The intention in estimating the minimal lifetime for comparison is to have an understanding of the network performance of TPSP in the worst case.

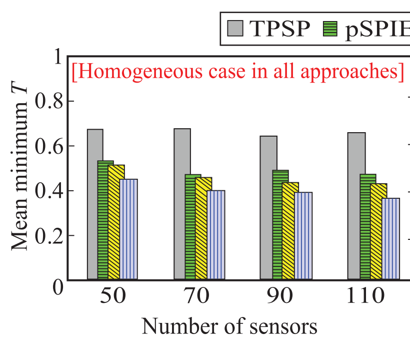
Figure 6(a) shows T of different approaches in the homogeneous WSN placement case, where we estimate the mean T. In Figure 6(b) and Figure 6(c), we compare the mean maximum T and mean minimum T, respectively. We can observe that, in the case of the mean minimum T, TPSP achieves longer lifetime (about 35% to 45%) than SPEM, UniRan, and pSPIEL. The mean maximum T is much longer (about 45% to 73%)



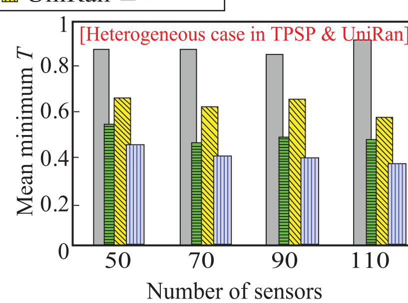
(a) network lifetime with a varying number of sensors (fault-free condition)



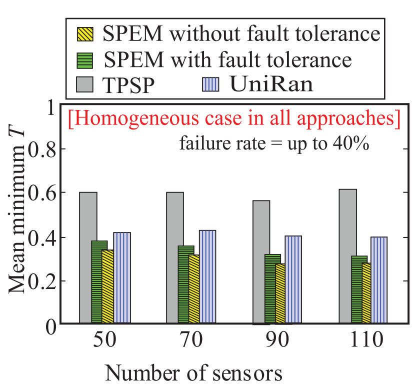
(b) network lifetime with a varying number of sensors (fault-free condition)



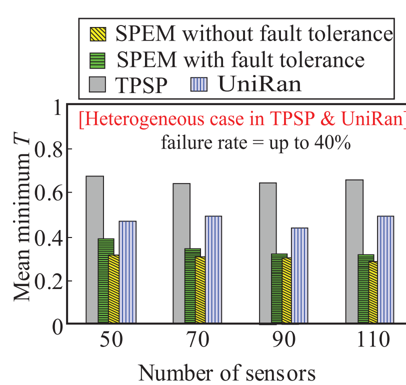
(c) network lifetime with a varying number of sensors (fault-free condition)



(d) network lifetime with a varying number of sensors (fault-free condition)

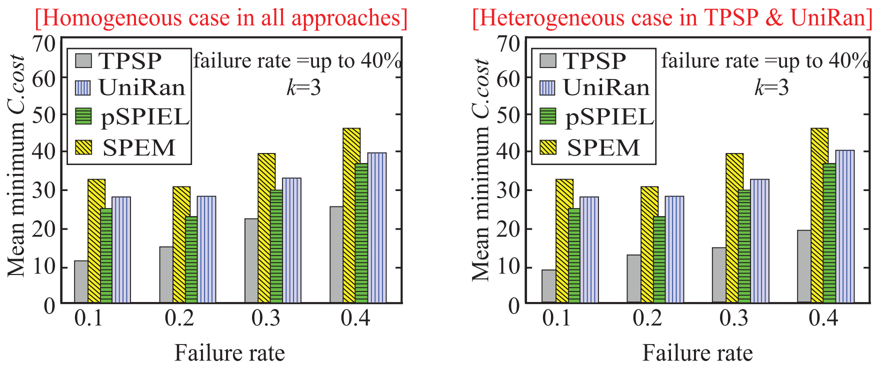


(e) network lifetime vs. the number of sensors under fault tolerance



(f) network lifetime vs. the number of sensors under fault tolerance

Fig. 6. Network performance achieved through simulations: (a) mean T under sensor fault-free condition; (b) mean maximum T under sensor fault-free condition; (c) mean minimum T under sensor fault-free condition; (d) mean minimum T under sensor fault-free condition (heterogeneous WSN placement case in TPSP and UniRan); (e) mean minimum T under fault-tolerance condition; (f) mean minimum T under fault-tolerance condition (heterogeneous WSN placement case in TPSP and UniRan).



(a) the communication cost in the network under sensor fault tolerance

(b) the communication cost in the network under sensor fault tolerance

Fig. 7. Network performance results achieved through simulations: (a) $C.cost$ under fault-tolerance condition in all approaches (homogenous WSN case all approaches); (b) $C.cost$ under fault-tolerance condition in all approaches (heterogeneous WSN case in TPSP and UniRan).

than all of the approaches. We decrease the communication load (thus communication cost) on sensors so as to reduce energy consumption. We ensure the placement of sensor nodes at the reliable and communication-efficient locations (similar to pSPIEL).

As the basic design of TPSP, the HNs (high-end nodes) are deployed with a higher amount of energy than the amount of energy given for the LNs. It is interesting to mention that, in the second simulation campaign, when the HNs are given double the energy and communication range of that the LNs, the amount of energy consumption by HNs did not cross the limit that was set to the LNs. However, we found that the HNs have a degree (average 7%) of higher energy consumption. This is because the HNs have additional tasks and have longer-distance communication than those of the LNs.

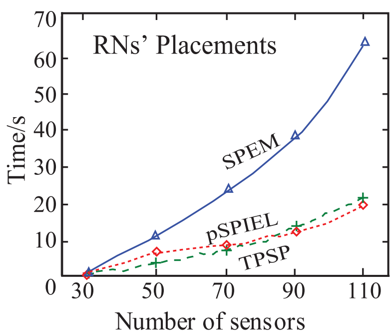
As the performance on the mean minimum T is analyzed in Figure 6(d), we can observe that T increases from 27.1% to 43.2% in TPSP. Compared to other approaches, T is 27.1% longer than UniRan, 36.3% longer than pSPIEL, and 43.2% longer than SPEM. When estimating the mean maximum T, we found that T in TPSP is 42.3% longer than UniRan, 54.6% longer than pSPIEL, and 72.2% longer than SPEM. The network lifetime depends on the sensor node duty cycle. We consider collecting continuous vibration monitoring data. If we scheduled the sensing unit at each individual sensor node to enable a duty cycle of 2% or less (for monitoring 10 minutes a day), a more maximized lifetime would be achieved.

The performances on T in Figure 6(e) and Figure 6(f) is in three measures: (i) SPEM (similar to EIM) without fault-tolerance support; (ii) SPEM with fault-tolerance support; (iii) TPSP (including the RN placement) and UniRan (including the relay node placement). Note that in the first two measures, we do not execute the third phase of TPSP, meaning that all of the sensors are placed in the second phase after HN placement. The injected failure rate is up to 40%. It is found that T (mean minimum) is affected under faults as N increases, as shown in both Figure 6(d) and Figure 6(e). Based on the results in Figure 6(e), we make an observation about the influence on T when the first sensor node fails. We found that T is affected by 0.23% in TPSP, by 0.51% in pSPIEL, by 0.76% in UniRan, and by 0.91% in SPEM. This implies that TPSP consumes lower energy consumption than other approaches in case of the first sensor node failure.

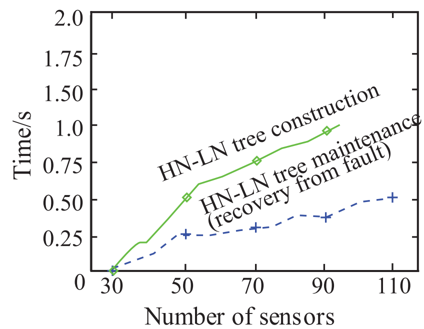
Figure 7(a) and Figure 7(b) plot $C.cost$ when the failure rate is up to 40%. We randomly remove 40% of the sensors. Here, $k = 3$. $C.cost$ in pSPIEL is much lower

Table II. Time Complexity on the Number of Nodes Placed and Network Maintenance

Algorithm	Nodes	Time
HNs Placement	$O(n_h d)$	$O({n_h}^M d)$
LNs Placement	$O(n_l)$	$O(n_l^3 M)$
Grouping and HN-LN tree	$O(n_l + n_h)$	$O((n_l + n_h)^2)$
RNs Placement	$O(n_r)$	$O((n_h + n_l) \cdot d)$
Connectivity Maintenance	n	$O(e(e + n))$



(a) the computation time as function of the number of sensors



(b) the computation time as function of the number of sensors

Fig. 8. Network performance results based on the placement algorithms attained through simulations: (a) required computation time for RNs' placements; (b) required computation time for sensor grouping into HN-LN tree and tree maintenance.

than SPEM and UniRan, while TPSP attains a low communication cost compared to all of them in case of the fault tolerance. When a node fails, the network is not separated; whether an RN continues working at the location or not, connectivity and coverage of the closest nodes in the corresponding HN-LN tree are improved. The whole WSN is not affected by the failure, that is, sensor nodes located in the specific substructure are involved in the improvement. The required amount of $C.cost$ in different approaches helps to observe the total amount of additional energy consumption.

Table II summarizes the assessments of time complexity and the number of sensors placed through different algorithms in TPSP. Figure 8(a) plots the computation time and the number of RNs deployed as functions of the total number of sensors. We observe that the results of evaluation of algorithms coincide with the analytical assessments shown in Table II. We see that the number of RNs has an asymptotic behavior as the number of sensors increases. This means that there is a saturation point in increasing the number of sensors when, for a new RN deployed beyond this point, there is always at least one route of already deployed LNs connecting the RN to the HN. Figure 8(b) shows the computation time of the HN-LN tree construction and the tree maintenance for sensor fault tolerance.

We next recover the first four mode shapes (Φ) with a total of (N=) 110 sensor locations, as shown in Figure 9 under the proposed TPSP approach. The datasets of GNTVT are used for this purpose. Although N is large, the adopted data can efficiently rebuild mode shapes of GNTVT because of the sensor placement method and the WSN architecture. The synchronization accuracy of the data is considered reliable [Araujo et al. 2012] and the obtained mode shapes by TPSP are close to the actual mode shapes of the GNTVT.

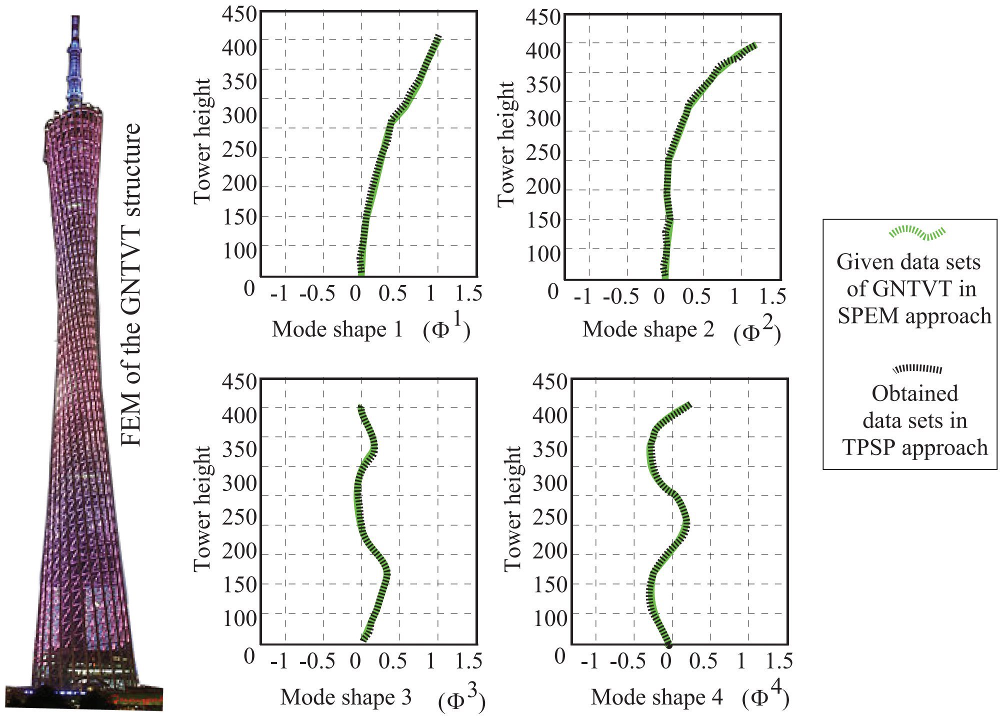


Fig. 9. The GNTVT structure and its four global mode shapes, identified by analyzing datasets in both SPEM and TPSP.

The preceding results hint that TPSP achieves almost the same performance in SHM as that achieved by CSE with EIM-based placement; besides, TPSP ensures the achievement of the multiple objectives in the WSN. By examining each sensor location state in the obtained mode shapes separately, the quality of SHM (e.g., damage detection) can be analyzed. In CSE, accurate mode shape identification specifies the high damage detection ability of an SHM system.

7. REAL EXPERIMENTS

7.1. Methodology

To validate the efficiency and effectiveness of the TPSP, we computed sensor placement for monitoring a real building structure: the *Lee Shao Kee (LSK)* tower, which is located at the Hong Kong PolyU campus (see Figure 10). Our objective in conducting the experiment is to observe cyber-physical aspects: real Φ identification on different sensor locations, E_j, $C.cost$, and T.

As a baseline deployment, we select 22 locations that seem to capture the overall vibrational characteristics. To address the generally high requirements of SHM applications, we design a particular type of wireless sensor nodes called *SHM motes*. Each SHM mote is tailored by an Imote2 platform [Crossbow Technology 2007], a sensor board, and a radio-triggered wakeup, as shown in Figure 11. The LSK tower has 14 floors, and the SHM motes are deployed on the floors to monitor the structure's horizontal accelerations under ambient vibrations. We have conducted the experiments on two different days.

We provide three `HN`s with double the communication and processing power, while we provide 13 `LN`s and six `RN`s that are resource constrained. Note that the Imote2's

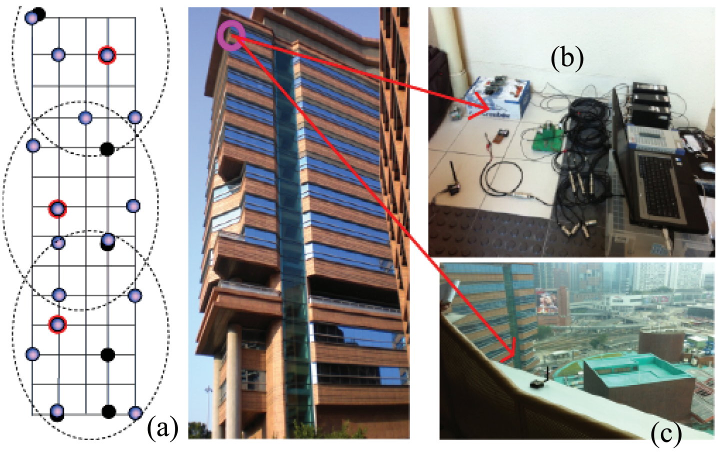


Fig. 10. Heterogeneous WSN deployment on the LSK building: (a) an example of the FEM model where three HNs (red circle), 13 LNs, and 6 RNs (black) are placed; (b) the sink location; (c) the placement of a sensor.

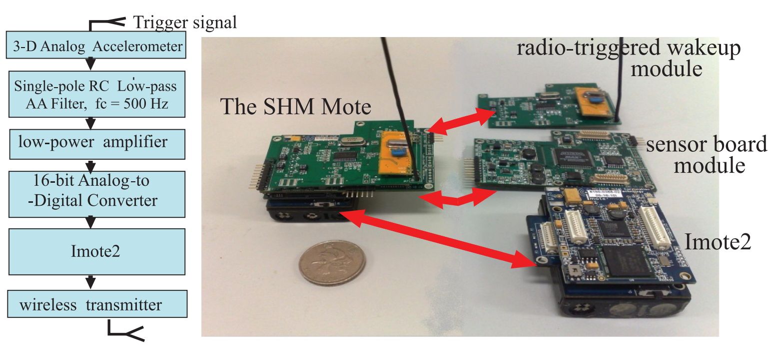


Fig. 11. The SHM mote integrated by Imote2 used in our experiments.

processor (PXA271) speed is scaled down [Crossbow Technology 2007] for both the HNs and LNs. Using the synchronized sensing middleware service [Rice and Spencer 2009], the Imote2 is programmed to capture acceleration responses of the structure. The LIS3L02DQ accelerometer [STMicroelectronics 2013] on the Imote2 sensor board applies an AA filter and yields digital outputs. The cutoff frequency of the AA filter is fixed with respect to a sampling frequency. The motes run the TinyOS [Levis 2006] and are configured to sample the accelerometers in a synchronized manner at a maximum frequency of 560 Hz. We observe the WSN with 22 sensor nodes plus the sink mote.

We find three optimal places for the three HNs. Then, we place 13 LNs on the structure. Finally, we find five places for five RNs. The placed motes connect themselves into three HN-LN trees (see Figure 10(a)) and form a WSN. Sensor faults are injected in the experiments by removing a connected LN from the fifth floor and disabling the sensing module on the second day. Meanwhile, the sensors that are placed on the third, fourth, fifth, sixth, and seventh floors are expected to improve their connectivity. Upon deployment, the motes can communicate with each other. To take advantage of the energy savings of the deep-sleep mode while still allowing the sink node access to the HNs and HNs to the LNs, a sleep/wake cycle technique, SnoozeAlarm, is adopted [Rice and Spencer 2009] with an improvement by which the motes are scheduled to switch sleep/wakeup. All of the motes operate their radios with a low duty cycle, similar to the setting in the simulations. The technique is implemented on the Imote2 motes. In an exception, if there is damage detected by a mote, the triggering wakeup module (as shown in Figure 11) is used to issue interrupt wakeup messages to the neighboring motes. A detailed description about the duty cycle and the module can be found in Appendix B. The time synchronization technique is similar to that used in the simulations.

7.2. Data Collection

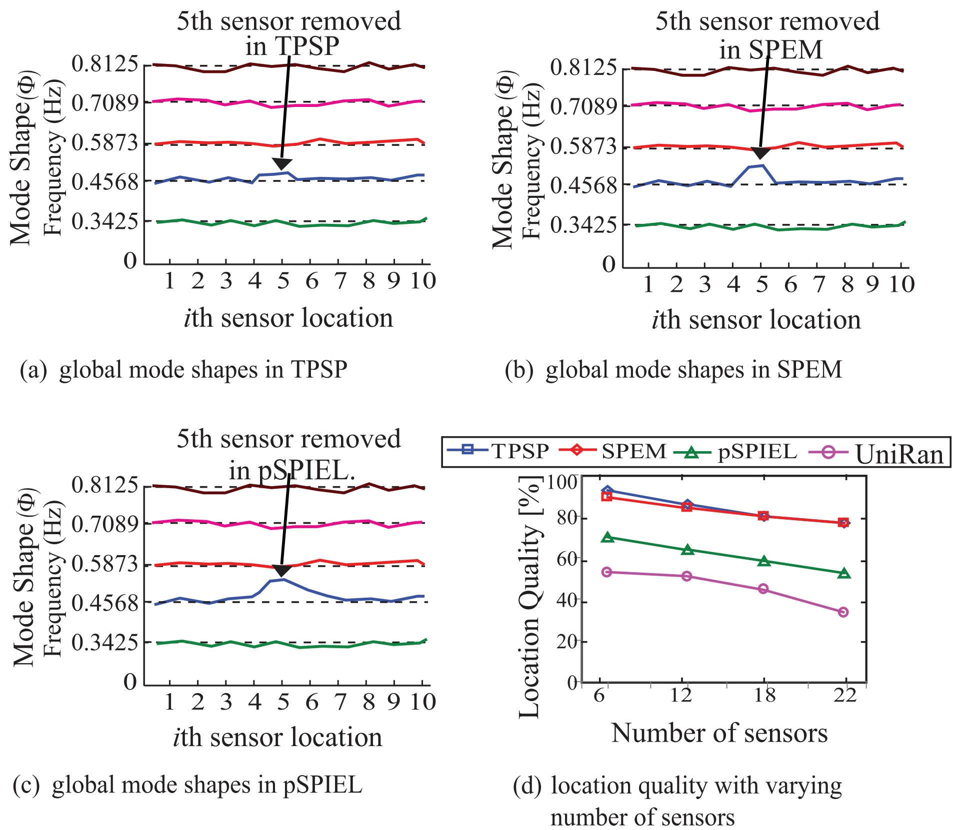
The data collection from the deployment is based on the TinyOS, which we extend to collect data in the vicinity of each sensor's location and the tree construction. Once per epoch, every sensor sends out a broadcast message containing its unique identifier and a decision message towards its `HN`s. We conducted the experiments on different days. On the first day of deployment, we first place the sensors according to the placement in TPSP, identify the location of each, and mark them. All of the sensors are allowed to collect structural measurements. Our intention is to create a reference mode shape for the vicinity of each sensor location. Each sensor computes the mode shape and stores it in its local memory, which remains in the memory after the SHM operation is over, and we use it for observation. A reference mode shape of the structure is identified based on the collected mode shape from each sensor on the first day, when there is no damage in the structure, or there is no sensor fault in the network.

On the second day, all of the sensors are deployed at the same places that were marked on the first day. Now, every sensor (`LN` or `RN`) is given the reference mode shape and is allowed to take measurements, and they refine a set of true mode shapes. They make a decision from the set and compare the final mode shapes with the reference one. If there is a significant discrepancy in the mode shape that they detect, they transmit the information to their `HN`s.

Upon message reception from all the `LN`s, the `HN` compiles a bitstring (implying from which `LN`s or `RN`s it has heard in the current epoch) and mode shape information. An `HN` then makes a decision based on the discrepancy in the mode shape. This transmission log information is then transmitted, along with its decision and all of the `LN`s' working state, via direct routing or through another `HN` to the sink. Having the true mode shape of structure, after analyzing, the sink is able to decide on the health state of the whole structure.

Since deployment has been conducted temporarily on different days in TPSP, after placing a sensor node at a high-quality location, we remove the sensor node at the end of the day. On the first day, we had to mark the location manually for further usage on the second day, and so on. There are reasons to mark the locations for future usage. In the simulations with the collected real datasets, it was not difficult to find a location (where a sensor node provides a high-quality indicator) and to identify the location for further sensor placement. However, in practice, it is hard to find a high-quality location and make an aggregated mode shape for a substructure, due to structural dynamics. Identifying the accurate location information for future usage by employing GPS (Geographical information system) is not always possible. It is both time and energy consuming to find the locations every day.

For example, in our initial deployment setup, we have tried to obtain a reference mode shape computed directly at the sink. It has taken more time (5 to 7 times longer) than a normal monitoring time window to transfer all of the samples at the sink (mainly due to wireless traffic and channel capacity constraints). It is important to note that there will be no need to mark the locations and place the sensor nodes at the same locations when the SHM end-user deploys a WSN system permanently.



(a) global mode shapes in TPSP

(b) global mode shapes in SPEM

(c) global mode shapes in pSPIEL

(d) location quality with varying number of sensors

Fig. 12. (a)–(c) Identified five global mode shapes with different frequencies in TPSP, in SPEM, and in pSPIEL, respectively; (d) performance on the location quality.

7.3. Experiment Results

Figures 12(a) to 12(c) demonstrate the identified first five mode shapes under the sensor fault injection in TPSP, SPEM, and pSPIEL, and reveal some remarkable changes that are affected by the fault. In the experiments, the mode shapes corresponding to the ambient vibration at frequency set [0.3Hz, 0.8Hz] are extracted in the three HNs and then are sent to the sink. The experiments calibrate the FEM of the structure with the collected real data. We found that there are four overlapping sensors between the first and second HNs. The results reveal that the TPSP can recover from the failed (fifth) sensor situation and get the mode shape of the location, while both SPEM and pSPIEL fail to capture full mode shape information. Although there is no RN available at the fifth location and no recovery mechanism for the situation in both SPEM and pSPIEL, mode shape identification of the fifth sensor location is guaranteed in TPSP, as shown in Figure 12(a).

Figure 12(d) demonstrates a performance comparison between all four approaches in terms of location quality for the 22 sensors, which contrasts the location quality during sensor placement. It can be seen that the quality for the remaining locations reduces as the number of sensor placements increases. Observing the results of location quality in Figure 12(d), pSPIEL provides better location quality than UniRan. Compared to others, the performance of TPSP ensures the achievement of almost the

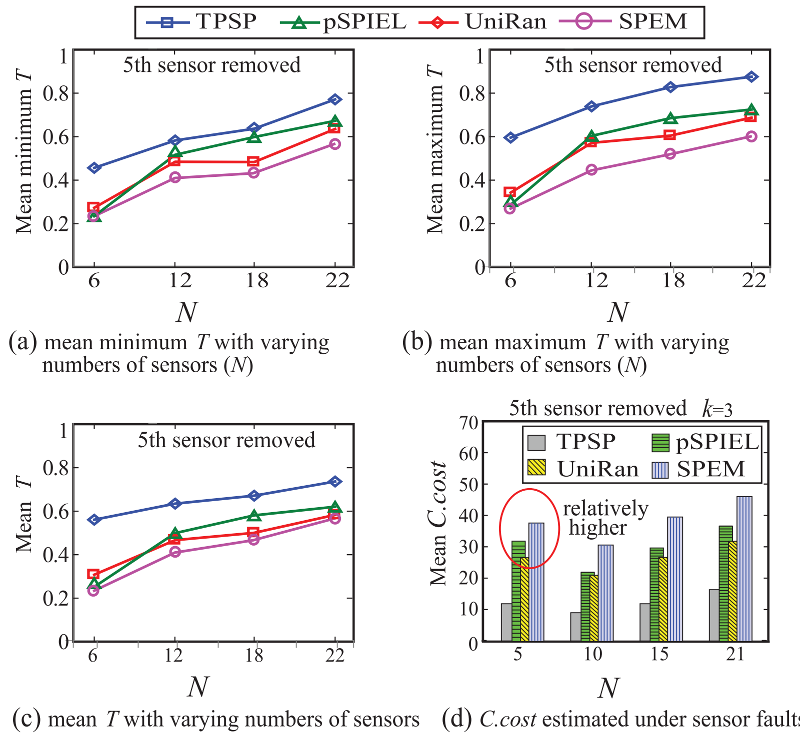


Fig. 13. Experimental results obtained through the WSN placement: (a) mean minimum T; (b) mean maximum T; (c) mean T; (d) mean communication cost.

same performance as the placement methods from CSE domains, which validates the correctness of our simulation results for the placement algorithms.

In Section 5.2, we have presented the performance of TPSP in terms of packet drop rates, and we compared the performance results of TPSP with other approaches. Here, we analyze the network lifetime (T) of TPSP and different approaches. As seen in Figure 13(a), the mean minimum T in TPSP is better than pSPIEL, while pSPIEL performs better than SPEM and UniRan. The mean minimum T in pSPIEL is closer to T in TPSP in 12-sensor and 18-sensor cases, though not in 6-sensor and 22-sensor cases.

Looking into the detail, the effect of the fifth sensor failure can be seen in the 6-sensor case. The neighboring `LN`s communicate to recover from the situation of the fifth sensor failure. This is because we have at least an `RN` sensor placed at the same location, and connectivity is maintained without degrading transmission energy. Similarly, the mean maximum T and mean T can be seen in Figures 13(b) and 13(c), respectively, where T is much larger in TPSP than in other approaches.

Figure 13(d) compares $C.cost$ of the four approaches in the presence of a sensor fault. As we can see, $C.cost$ is higher at the beginning; this is because the failed sensor belongs to the first `HN-LN` tree. $C.cost$ increases gradually as N increases afterward. Although there is no `RN` available at the fifth sensor location and no recovery mechanism for the situation in SPEM, UniRan, and pSPIEL, mode shape identification of the fifth sensor location is not guaranteed in all three approaches. TPSP can efficiently mitigate this situation.

8. CONCLUSIONS

Our study goes beyond the literature in at least three important aspects. First, we studied the sensor placement problem as a multiple-objective optimization problem for monitoring engineering structures, which provides new insights into the way we can get a distributed WSN-based SHM, and it addresses cyber-physical system issues. Second, we exhibited how the objectives of low communication, fault tolerance, and prolonged lifetime in a multidisciplinary WSN deployment approach can be achieved. Third, we evaluate the proposed approach via both simulations and a real implementation to validate its efficiency and effectiveness, where we ensure similar placement quality with that of engineering domains.

This work can be extended in two directions in the future. One direction is to develop a distributed and real-time data collection strategy for SHM based on the connectivity tree. Since SHM is a data-intensive application, modeling network data traffic is a difficult task. Another direction is to develop SHM-specific redundant sensor scheduling techniques, which will wake up one or more redundant sensors in the areas of interest (e.g., damaged area) in the case of a sensor fault/failure. This may help to meet both coverage and connectivity requirements in a WSN-based SHM system.

APPENDIXES

A. COMMUNICATION EFFICIENCY AND LINK RELIABILITY

Besides the energy required (modeled in Section 3.3) for communication between adjacent sensors, an important communication metric is the quality of a wireless link, which is typically characterized by the packet throughput, or alternately packet loss over the link [Gnawali et al. 2006]. Packet loss is dependent on distance $d_{u,v}$, with losses being higher for sensor nodes with larger separation or communication distance.

On the one hand, it is well known that a deployed network performs well in a controlled situation but poorly in practice, even at low data rates. In many monitoring environments, especially in different structural environments, links can be highly dynamic and bursty [Srinivasan et al. 2008; Munir et al. 2010]. Thus, during the data routing, the reliability requirement should be guaranteed.

On the other hand, in the case of SHM application, if a data packet transmitted by a node placed at an optimal location drops on the way, an important data (containing accumulated mode shape or a simplified decision on an extremity of a damage state) may be lost. Since the sensor placement for an SHM approach is characterized by the high location-quality indicator so as to fulfill the CSE domain requirements, the placement affects not only the routing protocols employed for data collection, but also the reliable collection of the transmitted data at the sink. The situation becomes more serious if an SHM end-user expects to collect all the recorded vibration signals (such a volume of data generated may require compression so as to reduce the amount of data transmitted [Ceriotti et al. 2009]).

We attempt to achieve communication efficiency and link reliability in two stages: during sensor placement and during network runtime. In the first stage (during the sensor placement at high-quality locations), we predict that the sensor placement has reliable communication links and the number of unnecessary retransmissions is minimized. We can calculate $R_{u,v}$ as the probability of the expected number of retransmissions between locations of any two sensors u and v, as suggested by Krause et al. [2011].

$$R_{u,v} = \rho \int_{\alpha} \frac{1}{\alpha_{u,v}} p(\alpha_{u,v}) d\alpha_{u,v}. \tag{5}$$

In (5), $\alpha_{u,v}$ is the probability for a successful packet transmission between the locations of any two sensors u and v, and $(\alpha_{u,v})^{-1}$ is the expected number of retransmissions, since the success packet transmission probabilities between any two locations depend on a probability distribution $P(\alpha_{u,v})$ with density $p(\alpha_{u,v})$ instead of a fixed value for $\alpha_{u,v}$ [Krause et al. 2011].

Using the formula in (5), we obtain the link quality denoted by $Q_{u,v}$ between any two sensor locations u and v. When analyzing $Q_{u,v}$ for sensor locations u and v, we can set $Q'_{u,v}$ as an acceptable (required) link quality, that is, we define $Q'_{u,v}$ as a threshold at a specified cutoff point for the link quality. If an obtained link quality between any two sensors u and v is above the acceptable link quality ($>Q'_{u,v}$), the link is considered a high-quality (or reliable) link.

The aforesaid model is suitable for our needs in deploying a WSN system for SHM, as it allows the estimation of the predictive energy cost and the communication cost of unsensed locations. The method used to find highly communication-efficient locations for sensor placement was very similar to the sensor placement on structures. This also ensures the quality of sensor placement on structures to some extent. However, there are still difficulties in applying the model during sensor placement on structures, considering distance as the parameter to determine link quality.

It is possible that, once a sensor node is placed with the high link quality, the deployed node may connect some of its neighboring nodes with the lowest communication cost. However, there are hurdles with (5) during sensor placement on a structure. The link quality can be *low* at some locations with *high*-quality indicators, that is, it is impossible to have reliable links in some locations in different structural environments, in at least in two instances: (i) there are physical structural constraints (see constraint C5 in Section 3.4), including obstacles (wall, pillars, piers); (ii) there is physical interference. On the other hand, the link quality may be low at some sensor locations with high location-quality indicators in some instances. We have an algorithm to handle these hurdles in Section 4.3.

In the second stage (during the network runtime), there exist significant network dynamics, such as communication link faults and congestion (though the placed sensor nodes are not mobile), making the static routing protocols unsuitable [Srinivasan et al. 2008]. Particularly, link faults in the trees of the network are highly possible in structural environments. The trees are built along with the sensor placement rooted at the HN(S), and then at the sink. We assume that some of the routing paths through the trees may be altered during monitoring operations. We validate the paths to account for the changes.

We apply the technique of datapath validation [Gnawali et al. 2006], which enables routing layers to remain efficient and reliable in highly dynamic topologies on many different link layers. If there are inconsistencies in the routing paths, the routing layer dynamically repairs the topology with a low overhead and forwards the packet normally. It does not require system-wide validation to recover from a routing path inconsistency; thus, it is suitable for the substructure-oriented monitoring.

B. LOW-POWER WAKEUP MECHANISM OF SENSOR NODES

To ensure an extended network lifetime, the network typically operates with low power consumption and, in many cases, does not require large radio transmission bandwidths. Low power consumption has to be achieved via the work/sleep mode along with a low duty cycle $< 0.1\% \sim 2\% >$, and low duty-cycle operation is usually feasible for WSNs since many SHM applications only require the collection of 5 to 10 minutes of data a day to reduce power consumption.

Although this duty-cycle setting is suitable in some cases (e.g., monitoring the states of industrial machine structures), this brings at least two concerns in monitoring some other structures (building, bridge, aircraft, etc.) that should be addressed.

First. It is hard to tune the duty-cycle interval in a balance between real-time functioning and energy saving. A short interval offers higher sensitivity, but consumes more resources, due to frequent sampling and transmissions. A long interval reserves more energy at the price of increased risk of missing important events and data. Complex environments and a variety of application scenarios make the tradeoff hard to balance.

Second. A fixed duty-cycle scheduler lacks situation awareness. A WSN may not be able to perform persistent monitoring with quick events identification and situation assessment on a fixed duty-cycle scheduler because some structural events (earthquake, cracks in a structure, gust of wind, hit by other events) may occur at any moment. Aircrafts, ships, vessels, etc., require continuous monitoring. If the monitoring is performed for such a short or long period a day, the WSN may fail to detect the events.

A sleep/wake-cycle technique called "SnoozeAlarm" is used by Rice and Spencer [2009] and Jang et al. [2010] in their SHM approaches can overcome the preceding concerns to some extent. SnoozeAlarm provides sleep-cycle functionality, which greatly reduces long-term energy consumption. Sensor nodes sleep for a period of time and then wake up for a relatively short period, during which they can interact with the network.

In the technique, some specifically designed wireless nodes, generally called *sentry nodes*, are deployed. They continuously monitor the events (the sensing unit is with a duty cycle of 100%) and send wakeup messages when an event is detected. The remaining ones are put into the SnoozeAlarm mode:they periodically wake up to listen for a while because they need to determine whether there are possible wakeup messages from the sentry nodes and to decide to work or to sleep accordingly.

However, the SnoozeAlarm also has some disadvantages from placement perspectives. (i) It cannot provide the coverage of some important locations, due to the duty cycle that two types of nodes wake up at different times. As a result, the important locations that are covered by one type of node, which are in sleep mode, are not monitored. (ii) The sentry nodes require continuous power supply. (iii) For monitoring purposes, it needs network-wide flooding, and the deployed sensor nodes are required to be within the single-hop communication range of at least one sentry node. Otherwise, routing paths between the nodes may be unavailable for the nodes, which will wake up in alternative times.

The TPSP can improve the disadvantages. Using the idea of the SnoozeAlarm and the vibration threshold (obtained by the signal processing algorithm used in Section 5.3), we develop a radio-triggered wakeup module [Liu et al. 2013] similar to the "radio-triggered circuit" used by Ansari et al. [2009]. The module shown in Figure 11 is connected to the Imote2 main board via an external pin of Imote2 (without requiring any extra interface). When a node's captured vibration parameters exceed the given vibration threshold (obtained by the signal processing algorithm used in Section 5.3), the module is enabled at once to generate interrupt wakeup messages to wake up the corresponding node once it makes a positive decision "damage," or collects a positive decision "damage." In other words, a node receives wireless packets sent from others that contain decisions similar to its own, or contain enough signal energy information in the case of raw data transmission. The energy consumption of the radio-trigger wakeup units is less than 1mA. More details about this radio-triggered wakeup mechanism can be seen in Liu et al. [2013].

In TPSP, HNs can work like the sentry nodes. Both HNs and LNs (including RNs) are enabled to work at the same time with minor modifications on the duty cycle. After collecting the data and making the decision, if there is a "no damage" state, the LNs (and RNs) should immediately enter into sleep mode. If there is damage, most of the locations in a substructure can be covered and monitored, since the two types of nodes can work. Both HNs and LNs are given limited battery power in TPSP. However, the HNs can be given continuous power supply, if one wishes.

On the one hand, to detect a structural event in realtime, the sensing unit of all of the sensors can be set to continuously collect vibration data, or can be set to be more frequently active than the radio units. The radio unit is set to work along with a duty cycle of 2%. On the other hand, due to having more communication tasks, the duty cycle (4%) of HNs' radio units is slightly longer than the duty cycle (2%) of LNs and RNs. By functioning at a low duty cycle, that is, the fraction of time that a node's radio is active/on, the node is able to save energy and consequently maximize its lifetime, while ensuring the quality of monitoring. Note that if both HNs and LNs have the same communication capabilities, the duty cycle can be set to the same value.

C. COMPUTATION OF E_J

According to the EIM, a mathematical mode shape of a structure denoted by Φ is based on excitation and is characterized with noise effects. Ambient or force excitation as input vibration is captured by a sensor.

Assume that there are M feasible locations on the structure and we need to find N locations. Each of the $N(<M)$ sensors can measure the vibration signals generated by the N locations. The vibration data can be represented by a column vector, and then the mth mode shape Φ^m measured by jth sensor can be seen as

$$\Phi^m = \left[\phi_1^m, \phi_2^m, \ldots, \phi_j^m\right]^T. \tag{6}$$

Φ^m conveys the individual contribution of the jth sensor location to the measurement data matrix X that is given by

$$X = \Phi^m \cdot q + \sigma, \tag{7}$$

which is comprised of the structural element displacements (i.e., response) that correspond to the sensor locations on the FEM model. The vector q represents the contribution of response that is estimated, and σ represents the measurement noise, as in Yi et al. [2011].

X is optimized as the best estimation of vector q is obtained. It can be obtained by minimizing the error covariance matrix, defined as follows [Meo and Zumpano 2005; Kammer 1990]

$$X' = E[(q - \hat{q})(q - \hat{q})^T] = [(\Phi^m)^T R^{-1} \Phi]^{-1} = Q^{-1}, \tag{8}$$

where $\hat{q}$ is the estimation of q, $E(\cdot)$ denotes the expectation operator, R is the covariance matrix of the noise, and Q is the Fisher Information Matrix (FIM). Q can be rewritten as follows.

$$Q = (\Phi^m)^T R^{-1} \Phi \tag{9}$$

In order to decide which sensor location leads to a higher signal strength, Q is used to determine the contribution of each sensor location to the measurement data matrix. In that case, Q is modified as

$$Q_T^j = Q - \left(\Phi_j{}^m\right)^T \Phi, \tag{10}$$

where $\Phi_j{}^m$ is the jth row of the mode shape associated with the jth candidate sensor location. The determinant of the new FIM (Q) is

$$\det\left(Q_T^j\right) = \det(Q)\det(1 - E_j), \tag{11}$$

where E_j is the EIM location-quality indicator that corresponds to the jth sensor location, defined as:

$$E_j = \left(\Phi_j{}^m\right)^T Q^{-1} \Phi_j. \tag{12}$$

D. MODE SHAPE NORMALIZATION

Before describing the mode shape normalization, we refer to Definition 3.2 and Appendix C for more details about the mode shape. Consider that the structure is covered by overlapping subnetworks. The HN has the set of mode shapes collected from all of its LNs' final mode shapes. After aggregation of the final mode shapes sent by the LNs within a subnetwork, a decision is made by at least an HN independently for each substructure.

Assuming that the boundary nodes located in a substructure under an HN are partially overlapped by the nodes located in s neighboring substructures, where $s = 2$ or more, the mode shapes from the boundary nodes may be influenced by the s substructures, hence, ϕ_s^m for the mth mode shape that is associated with s individual substructures. Mode shapes of any two adjacent substructures, Φ_i^m and Φ_j^m, respectively, are given as

$$\begin{array}{l} \Phi_i^m = \left[\phi_i^{n,1}, \phi_i^{n,2}, \ldots, \phi_i^{n,p}, \phi_i^{o,1}, \phi_i^{o,2}, \ldots, \phi_i^{o,q}\right] \\ \Phi_j^m = \left[\phi_j^{n,1}, \phi_j^{n,2}, \ldots, \phi_j^{n,p}, \phi_j^{o,1}, \phi_j^{o,2}, \ldots, \phi_j^{o,q}\right] \end{array}, \tag{13}$$

where n is the number of nonoverlapping sensors in the ith and jth substructures. o is the number of the overlapping sensors, and p and q are the number of overlapping and nonoverlapping sensors in the ith and jth substructures, respectively. Mode shapes from any two adjacent substructures can be rescaled to one, in terms of overlapping sensors, because any overlapping sensor nodes have the same values for different substructures. For example, the ith and jth substructures are normalized as

$$\Omega_i\left[\phi_i^{o,1}, \phi_i^{o,2}, \ldots, \phi_i^{o,q}\right] = \Omega_j\left[\phi_j^{o,1}, \phi_j^{o,2}, \ldots, \phi_j^{o,q}\right]. \tag{14}$$

Hence, the final mode shapes can be given by associating all of the mode shapes of s substructures as

$$\Phi_s^m = \bigcup_{i=1}^{s} \Omega_l \Phi_i^m. \tag{15}$$

In the presence of noise in the structural environment, the normalized solution in (13) for any $q > 1$ does not exist in practice. Thus, the normalization factor $\Omega_l(1, 2, \ldots n)$ is approximately determined, for example, as a solution in the least-square sense. Using the normalization factor R, the local mode shapes are scaled and assembled to obtain the global mode shape. At the overlapping nodes in the WSNs, the local mode shapes are averaged to obtain the associated values of the global mode shape.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for providing insightful comments and suggestions to improve the quality of the article, and to thank the shepherd Prof. Gian Pietro Picco for his guidance.

REFERENCES

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: A survey. *Comput. Netw.* 38, 4, 393–422.

J. Ansari, D. Pankin, and P. Mahonen. 2009. Radio-triggered wake-ups with addressing capabilities for extremely low power sensor network applications. *Int. J. Wirel. Inf. Netw.* 16, 3, 118–130.

K. Anshuman, J. Ratneshwar, W. Matthew, and J. Kerop. 2013. Damage detection in an experimental bridge model using Hilbert Huang transform of transient vibrations. *Struct. Control Health Monitor.* 20, 1, 1–15.

A. Araujo, J. Garca-Palacios, J. Blesa, F. Tirado, E. Romero, A. Samartn, and O. Nieto-Taladriz. 2012. Wireless measurement system for structural health monitoring with high time-synchronization accuracy. *IEEE Trans. Instrum. Meas.* 61, 3, 801–810.

S. Beygzadeh, E. Salajegheh, P. Torkzadeh, J. Salajegheh, and S. Naseralavi. 2013. Optimal sensor placement for damage detection based on a new geometrical viewpoint. *Int. J. Optim. Civil Engin.* 3, 1, 1–21.

M. Z. A. Bhuiyan, J. Cao, and G. Wang. 2012a. Deploying wireless sensor networks with fault tolerance for structural health monitoring. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'12)*. 194–202.

M. Z. A. Bhuiyan, J. Cao, G. Wang, and X. Liu. 2012b. Energy-efficient and fault-tolerant structural health monitoring in wireless sensor networks. In *Proceeding of the 31st International Symposium on Reliable Distributed Systems (SRDS'12)*. 301–310.

M. Z. A. Bhuiyan, G. Wang, and J. Cao. 2012c. Sensor placement with multiple objectives for structural health monitoring in WSNs. In *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC'12)*. 699–706.

M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu. 2013a. Energy and bandwidth-efficient wireless sensor networks for monitoring high-frequency events. In *Proceedings of the IEEE International Conference on Sensing and Communication, and Networking (SECON'13)*.

M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. WU. 2013b. Local monitoring and maintenance for operational wireless sensor networks. In *Proceedings of the 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA'13)*.

M. Bocca, L. Eriksson, A. Mahmood, R. Jntti, and J. Kullaa. 2011. A synchronized wireless sensor network for experimental modal analysis in structural health monitoring. *Comput.-Aid. Civil Infrastruct. Engin.* 26, 7, 483–499.

L. J. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus. 2010. Deploying sensor networks with guaranteed fault tolerance. *IEEE/ACM Trans. Netw.* 18, 1, 216–228.

M. Cardei, S. Yang, and J. Wu. 2008. Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 19, 4, 545–558.

M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. 2009. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN'09)*. 277–288.

X. Chang, R. Tan, G. Xing, Z. Yuan, C. Lu, Y. Chen, and Y. Yang. 2011. Sensor placement algorithms for fusion-based surveillance networks. *IEEE Trans. Parallel Distrib. Syst.* 22, 8, 1407–1414.

P. Cheng, C. N. Chuah, and X. Liu. 2004. Energy-aware node placement in wireless sensor networks. In *Proceedings of the IEEE International Conference on Global Communications (GLOBECOM'04)*. 3210–3214.

Crossbow Technology. 2007. Imote2 hardware reference manual. http://web.univ-pau.fr/~cpham/ENSEIGNEMENT/PAUUPPA/RESAM2/DOC/Imote2_Hardware_Reference_Manual.pdf.

C. E. Devore. 2013. Damage detection using substructure identification. M.S. thesis, University of Southern California.

C. R. Farrar and K. Worden. 2012. *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley and Sons Ltd, Chichester, UK.

O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. 2009. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*. 1–14.

G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke. 2013. Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks. http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.30.

G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke. 2012. A holistic approach to decentralized structural damage localization using wireless sensor networks. *Comput. Comm.* 36, 1, 29–41.

D. A. Huffman, 1952. A method for the construction of minimum-redundancy codes. *Proc. IRE* 40, 9, 1098–1101.

H. S. Jang, Mechitov S. Cho JO, J. A. Rice K., S.-H. Sim, H.-J. Jung, C.-B. Yun, F. Billie, J. Spencer, and G. Agha. 2010. Structural health monitoring of a cable-stayed bridge using smart sensor technology: Deployment and evaluation. *Smart Struct. Syst.* 6, 5, 439–459.

A. Jindal and M. Liu. 2012. Networked computing in wireless sensor networks for structural health monitoring. *IEEE/ACM Trans. Netw.* 20, 4, 1203–1216.

D. C. Kammer. 1990. Sensor placement for on-orbit modal identification and correlation of large space structures. In *Proceedings of the American Control Conference*. 2984–2990.

A. Kashyap, S. Khuller, and M. Shayman. 2011. Relay placement for fault tolerance in wireless networks in higher dimensions. *Comput. Geom. Theory Appl.* 44, 2011, 206–215.

S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. 2007. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN'07)*. 254–263.

A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. 2011. Robust sensor placements at informative and communication-efficient locations. *ACM Trans. Sensor Netw.* 7, 4, 1–33.

P. Levis. 2006. TinyOS/nesC programming reference manual. www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf.

B. LI, Z. Sun, K. Mechitov, C. Lu, S. J. Dyke, G. Agha, and Spencer. B. F. 2013. Realistic case studies of wireless structural control. In *Proceedings ACM/IEEE 4th International Conference on CyberPhysical Systems (ICCPS'13)*.

B. LI, D. Wang, F. Wang, and Y. Q. NI. 2010. High quality sensor placement for SHM systems: Refocusing on application demands. In *Proceedings of the Conference on Computer Communications (INFOCOM'10)*. 650–658.

L. E. Linderman, J. A. Rice, S. Barot, B. F. Spencer, and J. Bernhard. T. 2010. Characterization of wireless smart sensor performance. Tech. rep .021, Newmark Structural Engineering Laboratory, University of Illinois at Urbana-Champaign, Champaign, Illinois.

X. Liu, J. Cao, M. Z. A. Bhuiyan, S. Lai, H. Wu, and G. Wang. 2011a. Fault tolerant WSN-based structural health monitoring. In *Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. 37–48.

X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. Xu. 2011b. Energy efficient clustering for WSN-based structural health monitoring. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'11)*. 2768–2776.

X. Liu, J. Cao, and S. Tang. 2013. Enabling fast and reliable network-wide event-triggered wakeup in WSNS. In *Proceedings of the IEEE 24th Real-Time Systems Symposium (RTSS'13)*.

M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. 2004. The flooding time synchronization protocol. In *Proceedings of the 2^{nd} International Conference on Embedded Networked Sensor Systems (SenSys'04)*. 39–49.

M. Meo, and G. Zumpano. 2005. On the optimal sensor placement techniques for a bridge structure. *Engin. Struct.* 27, 10, 1488–1497.

S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse. 2010. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'10)*. 303–314.

Y. Ni, H. Zhou, K. Chan, and J. Ko. 2008. Modal flexibility analysis of cable-stayed ting kau bridge for damage identification. *Comput.-Aid. Civil Infrastruct. Engin.* 23, 3, 223–236.

Y. Q. Ni, Y. Xia, W. Y. Liao, and J. M. Ko. 2009. Technology innovation in developing the structural health monitoring system for Guangzhou new TV tower. *Struct. Control Health Monit.* 16, 1, 73–98.

P. Nie and B. Li. 2011. A cluster-based data aggregation architecture in WSN for structural health monitoring. In *Proceeding of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC'11)*. 546–552.

S. Olariu and I. Stojmenovic. 2006. Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'06)*. 1–12.

C. A. Peckens and J. P. Lynch. 2013. Embedded linear classifiers on wireless sensor networks for damage detection. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*.

J. A. Rice and B. F. Spencer. 2009. Flexible smart sensor framework for autonomous full-scale structural health monitoring. Tech. rep. 018, Newmark Structural Engineering Laboratory, University of Illinois at Urbana-Champaign, Champaign, Illinois.

S.-H. Sima, J. F. Carbonell-Mrquezb, B. F. S., and Joa, H. 2011. Decentralized random decrement technique for efficient data aggregation and system identification in wireless smart sensor networks. *Probab. Engin. Mechan.* 26, 1, 81–91.

S. Singh, M. Woo, and C. Raghavendra. 1998. Power-aware routing in mobile ad hoc networks. In *Proceedings of the ACM/IEEE Annual International Conference on Mobile Computing and Networking (MobiCom'98)*. 181–190.

K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. 2008. The β-factor: Measuring wireless link burstiness. In *Proceedings of the 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys'08)*. 29–42.

STMicroelectronics. 2013. http://www.st.com/internet/com/home/home.jsp.

F. Wang, D. Wang, and J. Liu. 2011. Traffic-aware relay node deployment: Maximizing lifetime for data collection in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 22, 8, 1415–1423.

G. Wang, M. Z. A. Bhuiyan, and Z. Li. 2010. Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks. *Concurr. Comput. Pract. Exper.* 22, 4, 518–537.

Q. Wang, M. Hempstead, and W. Yang. 2006. A realistic power consumption model for wireless sensor network devices. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*. 286–295.

Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng. 2008. Efficient placement and dispatch of sensors in a wireless sensor network. *IEEE Trans. Mobile Comput.* 7, 2, 162–174.

S. Weng, Y. Xia, Y.-L. Xu, and H.-P. Zhu. 2011. Substructure based approach to finite element model updating. *Comput. Struct.* 89, 2011, 772–782.

M. Whelan and C. Hietbrink. 2012. Wireless sensor network deployments for structural identification in a tied arch bridge. In *NDE/NDT for Highways and Bridges: Structural Materials Technology*.

S. Wijetunge, U. Gunawardana, and R. Liyanapathirana. 2009.Wireless sensor networks for structural health monitoring: Considerations for communication protocol design. In *Proceedings of the IEEE International Conference on Telecommunications (ICT)*. 694–699.

J. Wu. 2005. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*. Auerba

Z. Xing and A. Mita. 2012. A substructure approach to local damage detection of shear structure. *Struct. Control Health Monit.* 19, 2, 309–318.

K. Xu, H. Hassanein, G. Takahara, and Q. Wang. 2010. Relay node deployment strategies in heterogeneous wireless sensor networks. *IEEE Trans. Mobile Comput.* 9, 2, 145–159.

G. Xue, X. Fang, S. Misra, D. Yang, and J. Zhang. 2012. Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations. *IEEE Trans. Mobile Comput.* 11, 8, 1399–1411.

T.-H. Yi, H.-N. Li, and M. Gu. 2011. Optimal sensor placement for structural health monitoring based on multiple optimization strategies. *Struct. Des. Tall Special Buildings* 20, 7, 881–900.

Received September 2012; revised September 2013; accepted September 2013