EleSense: Elevator-Assisted Wireless Sensor Data Collection for High-Rise Structure Monitoring

Feng Wang*
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
Email: fwa1@cs.sfu.ca

Dan Wang†
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong
Email: csdwang@comp.polyu.edu.hk

Jiangchuan Liu*
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
Email: jcliu@cs.sfu.ca

***Abstract*—Wireless sensor networks have been widely suggested to be used in Cyber-Physical Systems for Structural Health Monitoring. However, for nowadays high-rise structures (e.g., the Guangzhou New TV Tower, peaking at $600m$ above ground), the extensive vertical dimension creates enormous challenges toward sensor data collection, beyond those addressed in state-of-the-art mote-like systems. One example is the data transmission from the sensor nodes to the base station. Given the long span of the civil structures, neither a strategy of long-range one-hop data transmission nor short-range hop-by-hop communication is cost-efficient. In this paper, we propose *EleSense*, a novel high-rise structure monitoring framework that uses elevators to assist data collection. In EleSense, an elevator is attached with the base station and collects data when it moves to serve passengers; as such, the communication distance can be effectively reduced. To maximize the benefit, we formulate the problem as a cross-layer optimization problem and propose a centralized algorithm to solve it optimally. We further propose a distributed implementation to accommodate the hardware capability of sensor nodes and address other practical issues. Through extensive simulations, we show that EleSense has achieved a significant throughput gain over the case without elevators and a straightforward 802.11 MAC scheme without the cross-layer optimization. Moreover, EleSense can greatly reduce the communication costs while maintaining good fairness and reliability. We also conduct a case study with real experiments and data sets on the Guangzhou New TV Tower, which further validates the effectiveness of our EleSense.**

I. INTRODUCTION

In recent years, wireless sensor networks (WSNs) have been widely suggested to be used in Cyber-Physical Systems (CPS), as WSNs can enable timely and efficient interactions between the cyber and physical worlds in many CPS applications. One typical application among them is Structural Health Monitoring (SHM) [29][11][4][8][3][13][19], where diverse sensor nodes are deployed on a structure, collecting ambient data such as temperature, strain and acceleration from various locations and reporting them to a central controller (the *base station*) for further processing, diagnosing and decision-making. Fig. 1 shows the SHM system deployed on the Guangzhou New TV Tower (GNTVT) in Guangzhou, China, a project in which we have participated.[1] Even during the construction phase, the tower has already been equipped with vibrating wire strain gauge sensors and temperature sensors (Fig. 1a) to monitor its construction status. After starting to fully operate in November 2010, more advanced sensing devices such as accelerometers and corrosion sensors are further deployed on the tower (Fig. 1b) to monitor its operation and service. As the world's tallest TV Tower, the GNTVT peaks at $600m$ above the ground. Although its horizontal dimension is similar to a normal building (varying from $50m \times 80m$ to $20.65m \times 27.5m$ at different floors), its pathological vertical dimension creates enormous challenges toward sensor data collection, beyond those addressed in state-of-the-art mote-like systems. One example is transmitting data from all sensor nodes to the base station. As data aggregation in SHM is not possible at the current stage [13], given the long span of the high-rise structures, neither a strategy of long-range one-hop data transmission (partially adopted by the GNTVT in its early stage) nor short-range hop-by-hop communication is cost-efficient. Recently there are studies that combine these two strategies by adjusting wireless communication ranges and/or adding more relay nodes for a more efficient system [28][9][24]. However, the intrinsic difficulty remains, i.e., the larger the structure, the longer the distances from the sensors to the base station. This has prohibited the GNTVT to install and harvest the benefit of a full-range wireless sensor system.

On the other hand, we note that the heights of these structures also make elevators indispensable in general. We thus propose *EleSense*, a novel high-rise structure monitoring framework to exploit elevators. In EleSense, an elevator is attached with the base station and collects data when it moves across different floors to serve passengers. As such, communication distances between sensors and the base station can be greatly reduced and the traffic relaying can be effectively balanced. Yet, to achieve optimal performance, there remain a series of theoretical and practical issues to be addressed. In particular, different from traditional mobile base stations that are fully controlled by the data collection applications, when and where an elevator would move or stop depend on its passengers, whose preferences can be in great variances and hardly be predicted and controlled. As a result, the sensor nodes at different floors may experience various capabilities

---

*Feng Wang and Jiangchuan Liu's research is supported by a Canadian NSERC Discovery Grant, a Discovery Accelerator Supplements Award, an NSERC Engage Grant, a MITACS Project Grant, and a China NSFC Major Program of International Cooperation Grant (61120106008).

†Dan Wang's work was supported by Hong Kong PolyU/A-PJ19, A-PK95, A-PL23, 4-BC01, 4-BC02, 4-BC03, and RGC/GRF PolyU 5305/08E.

[1] http://www.cse.polyu.edu.hk/benchmark/.

in transmitting data to the base station, making fairness and rate control difficult to achieve, which is further complicated by the dynamic interferences and collisions with a base station moving in real-time, and the limited power of wireless sensor nodes.

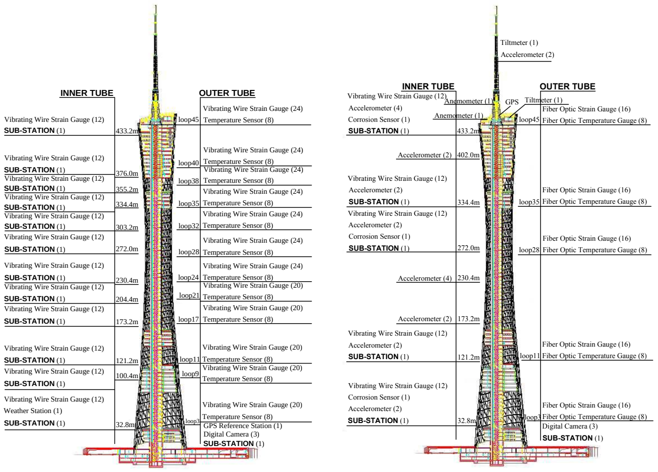
Being the very first paper to tackle these challenges, we strive to provide fundamental understandings on the practical feasibility and theoretical constraints. As both interference and data routing issues are involved, we take a cross-layer design approach and present a mathematical abstraction of the high-rise structure data collection problem as a joint optimization problem among link scheduling, packet routing and end-to-end delivery. We show that the centralized version of the problem can be solved optimally through dynamic programming, which provides a valuable benchmark and motivates a distributed implementation that accommodates the hardware limits of state-of-the-art sensor nodes and other practical issues.

We evaluate EleSense through both simulations in ns-2 and a case study with real experiments and data sets on the GNTVT. The results show that EleSense has a throughput gain of 30.7% to 212.7% over the case without elevators. We also observe a gain of 40.9% to 423.2% over a straightforward 802.11 MAC scheme without the cross-layer optimization. Moreover, EleSense can significantly reduce the communication costs while maintaining good fairness and reliability.

The remainder of this paper is organized as follows. We first discuss the background and challenges of high-rise structure monitoring in Section II, and then formulate the high-rise structure monitoring problem as a cross-layer optimization problem in Section III. In Section IV, we propose a centralized optimal solution to solve this problem, followed by a practical distributed implementation presented in Section V. We evaluate EleSense with extensive ns-2 simulations in Section VI, and the results are further confirmed in Section VII by a case study with real experiments and data sets on the GNTVT. Section VIII reviews the related work. Finally, we conclude this paper and offer some future directions in Section IX.

II. BACKGROUND AND CHALLENGES

In SHM applications, the sensors are deployed on critical locations that are of civil importance and periodically sample the data. For high-rise structures, a commonly adopted data collection strategy is to assign a representative node (e.g., sub-station in Fig. 1) on each floor to collect all the data from the sensors on this floor. These representative nodes then transmit the data back to the base station located at the foot of the structure. Conventionally, data transmission is carried out by wires. For a life-long monitoring system, a wire-dominated system may still be a reasonable choice. In this paper, we focus on the short term (weeks or months) evaluation of the structural health [13]. In this scenario, sensor networks are quickly deployed to acquire the data necessary for calibrating the structural health models for the civil engineering science. Based on the calibration, the deployment may be adjusted and new data collections may be issued for further verification and calibration until it meets the civil engineering requirements.



(a) (b)

Fig. 1: Sensor layout on the Guangzhou New TV Tower: (a) In-construction monitoring; (b) In-service monitoring.

Such an iterative procedure is often started even during the construction phase of a structure and makes a wired system introduce huge deployment costs. Another major headache is the in-construction monitoring, where the wires can be easily damaged by the hammers and drills during the structure construction. Wireless systems are thus welcomed for these situations especially considering the recent development of the state-of-the-art sensor systems.

There are two possible data collection strategies for wireless communication systems, namely, short-range hop-by-hop routing and long-range single-hop transmission. Long-range single-hop transmission (partially adopted by the GNTVT in its early stage) is costly in communication devices and suffers greatly if the energy supply is limited or difficult to obtain. Hop-by-hop routing will put high burden to the nodes that are close to the base station as they need to relay large amount of data[2], which may cause severe interferences and collisions at these nodes as well as unbalanced energy consumption that may shorten the network lifetime. Although carefully combining these two strategies may alleviate these problems; the intrinsic difficulty still remains, i.e., the larger the structure, the longer the distances from the sensors to the base station.

To this end, we propose EleSense as a generic framework for high-rise structure monitoring. In EleSense, the base station is installed on an elevator used by the structure. As passengers go to different floors, the base station moves with the elevator and collects data packets from the sensor nodes that it passes by. Nevertheless, there are still many challenges to be addressed. First, we cannot control the elevator; as such, the sensor nodes cannot always wait for the elevator to come. Though currently we target on the SHM applications not requesting for real-time data collection, and sensing data can be stored at the local external flash for later transmissions, collecting data as fast as possible is still welcome in general. Thus, the

---

[2] Note that to the best of our knowledge, in civil applications, data aggregation in the intermediate nodes is not practical for the time being (for background on civil data evaluation, one may refer to [13]).

sensor nodes have to decide whether to route the packets by neighboring nodes or wait for the base station arrival for direct transmissions. Also, when sending data packets, a sensor node must carefully schedule its transmissions to avoid wireless interferences and collisions. In addition, when a packet is relayed by a neighboring node, the queuing buffer limitation at that node should also be considered. In next section, we will formally model the high-rise structure monitoring problem by taking these issues into account.

III. PROBLEM STATEMENT

Consider that there are n floors in a given high-rise structure. For ease of exposition, we assume that there is one sensor node on each floor. Denote them as $s_1, s_2, \dots, s_n$. Let s_0 be the base station attached on the elevator. Let $l_{x,y}$ denote the directed link from node x to node y if packets can be transmitted along them, where $x, y \in \{s_0, s_1, \dots, s_n\}$. Because of the pathological extensions along the vertical dimension of high-rise structures, the link connection between a node and the base station may be unavailable when the base station moves far away with the elevator. We thus define $A(l_{x,s_0}, t)$ and let $A(l_{x,s_0}, t) = 1$ denote that at time t, the link connection between node x and the base station is available and otherwise $A(l_{x,s_0}, t) = 0$. Due to interferences and collisions, some links may not transmit data simultaneously. We define an interference matrix, shortened as IM, and let $IM(l_{x_1,y_1}, l_{x_2,y_2}) = 1$ if two links l_{x_1,y_1} and l_{x_2,y_2} interfere with each other when transmitting simultaneously, otherwise $IM(l_{x_1,y_1}, l_{x_2,y_2}) = 0$. We assume that all data packets have the same length and define a *time unit* as the minimum time for a link to be activated to reliably transmit a data packet. Sensor nodes are synchronized by a synchronization algorithm such as [14][18]. As data are often collected in a round by round manner in SHM applications, we let the data generated at node s_i be r_i packets per round, for $i = 1, 2, \dots, n$. Assume that each sensor node can buffer at most B extra data packets in its queuing buffer. Let t_0 denote the time that a data collection round starts.

The high-rise structure monitoring problem thus can be formulated as a cross-layer optimization problem to find a *link-activation schedule* $S = \{(l_{x_1,y_1}, t_1), (l_{x_2,y_2}, t_2), \dots, (l_{x_k,y_k}, t_k)\}$, $t_0 \le t_1 \le \dots \le t_k$, subjecting to the following constraints:

(1) *Link Availability Constraint*:

$$\forall (l_{x,y}, t) \in S,$$

$$\text{if } y = s_0 \text{, then } A(l_{x,y}, t) = 1 \ ;$$

(2) *Link Interference Constraint*:

$$\forall (l_{x_i,y_i}, t_i), (l_{x_j,y_j}, t_j) \in S,$$

$$\text{if } t_i = t_j \text{, then } IM(l_{x_i,y_i}, l_{x_j,y_j}) = 0 \ ;$$

(3) *Packet Transmission Constraint*:

$$\forall t \in [t_0, t_k], i = 1, 2, \dots, n,$$

$$r_i + \sum_{(l_{x_j,y_j}, t_j) \in S} I_{[y_j = s_i, t_j \le t]} \ge \sum_{(l_{x_j,y_j}, t_j) \in S} I_{[x_j = s_i, t_j \le t]} \ ;$$

(4) *Packet Buffering Constraint*:

$$\forall t \in [t_0, t_k], i = 1, 2, \dots, n,$$

$$\sum_{(l_{x_j,y_j}, t_j) \in S} I_{[y_j = s_i, t_j \le t]} - \sum_{(l_{x_j,y_j}, t_j) \in S} I_{[x_j = s_i, t_j \le t]} \le B \ ;$$

(5) *Traffic Source Constraint*:

$$\forall i = 1, 2, \dots, n,$$

$$r_i + \sum_{(l_{x_j,y_j}, t_j) \in S} I_{[y_j = s_i]} = \sum_{(l_{x_j,y_j}, t_j) \in S} I_{[x_j = s_i]} \ ;$$

(6) *Traffic Destination Constraint*:

$$\sum_{(l_{x_j,y_j}, t_j) \in S} I_{[y_j = s_0]} = \sum_{i=1}^{n} r_i \ ;$$

where $I_{[\cdot]}$ is the indicator function. The first two constraints are the requirements from the link layer. Since the base station moves with the elevator, (1) demands that the links to the base station must be available when being activated; (2) denotes that two simultaneously activated links must not interfere with each other. The next two constraints take the consideration of packet routing. Given any time instance, (3) implies that a node cannot deliver more packets than it has; and (4) indicates that a node cannot buffer packets more than its queuing buffer capacity (except for the base station). The last two constraints focus on end-to-end traffic, which follows that when the data collection round finishes, each node must send out all its data and the base station must receive all of them.

The objective function is thus to minimize $f(|S|, t_k - t_0)$, a function of the total number of transmissions ($|S|$) and the total latency ($t_k - t_0$), which is in general specified by the target application. In this paper, we will focus on a commonly-used linear combination, $f(|S|, t_k - t_0) = p|S| + q(t_k - t_0)$. By assigning different weights (p, q), it covers the demands from a broad spectrum of data collection applications. For example, if the collected data are about an emergency event, a small p plus a large q will ensure that the data packets are delivered to the base station in real-time, though possibly with more hop-by-hop transmissions. On the other hand, if the data are non-urgent while the transmissions (and thus the energy consumption) are the major concerns, a large p plus small q will work well to reduce the number of transmissions and save the energy costs. It is worth noting that EleSense can be easily extended to a hierarchial architecture (as the example of the GNTVT) where sensor nodes on the same floor form a cluster with sensing data sent to the representative node (sub-station) by local communications. Our analysis and algorithms below can be easily adapted to such situations.

IV. CENTRALIZED OPTIMAL SOLUTION

We first transform the centralized version of our problem into a shortest path problem in a time-state graph. Assuming that all the elevator movements are known, this graph problem is solvable through a dynamic programming algorithm. This provides us an understanding on the intrinsic complexity of the problem. Its design principle also motivates the distributed implementation to be presented in the next section.

A. Time-State Graph

We construct a directed graph $G(V, E)$, which we call a *time-state graph*. In this graph, vertices are organized in two dimensions, indexed by time (along the row direction) and state (along the column direction), respectively. Let $M = (m_0, m_1, \ldots, m_n)$ be a state, where node s_i has m_i packets, for $i = 0, 1, \ldots, n$. A vertex $v_{M,t}$ represents at time t the number of data packets at each node is given by state M. Note that the first row (with same state $(0, r_1, \ldots, r_n)$ but different time index) indicates that the base station s_0 has no packet and each s_i has r_i packets to deliver. The last row (with same state $(\sum_{i=1}^{n} r_i, 0, 0, \ldots, 0)$ but different time index) indicates that at a certain time, s_0 has all the packets collected.

There are two kinds of edges in the graph, referred to as *time edges* and *transmission edges*, respectively. A time edge connects two neighboring vertices along a row, from the earlier to the later. It corresponds to the case that no node transmits any packet at a time t, and the same state is thus inherited by the next time slot. A transmission edge, on the other hand, corresponds to link-activation events. Specifically, a transmission edge $(v_{M,t}, v_{M',t'})$ indicates that the network state changes from M at time t to M' at time $t' = t + 1$, by some link activations for data transmissions at time t (which must follow the constraints in Section III). We also set a series of special transmission edges $(v_{M,t}, v_{(\sum_{i=1}^{n} r_i, 0, 0, \ldots, 0), t})$, corresponding to the transmission edges directing to the last row, where all data have been collected and no transmission is necessary afterwards.

B. Equivalent Problem: Last Row Shortest Path

The time-state graph can be naturally correlated to our high-rise structure monitoring problem: Each link-activation schedule corresponds to a path from $v_{(0,r_1,\ldots,r_n),t_0}$ to a vertex in the last row, and vice versa. For the objective function $f(|S|, t_k - t_0) = p|S| + q(t_k - t_0)$, we assign weight q to each time edge since a delay of one time unit is incurred, and weight $(p|\mathbb{T}| + q(t' - t))$ to a transmission edge from $v_{M,t}$ to $v_{M',t'}$, where $\mathbb{T}$ is the set of activated links at time t that deliver data packets and change the state from M to M'. Our problem is then translated into the shortest path problem from $v_{(0,r_1,\ldots,r_n),t_0}$ to a last-row vertex in the weighted graph.

Let $W(v_{M,t}, v_{M',t'})$ denote the weight of the edge from $v_{M,t}$ to $v_{M',t'}$, and $W(v_{M,t}, v_{M',t'}) = \infty$ if the edge does not exist. Also let $F(v_{M',t'})$ be the total weight of the shortest path from vertex $v_{(0,r_1,\ldots,r_n),t_0}$ to $v_{M',t'}$. We have the following recurrence relation:

$$F(v_{M',t'}) = \min_{v_{M,t}} \big(F(v_{M,t}) + W(v_{M,t}, v_{M',t'})\big),$$

where $t = t'$ for $M \neq M'$ with $M' = (\sum_{i=1}^{n} r_i, 0, \ldots, 0)$; otherwise $t = t' - 1$. For boundaries, we have $F(v_{(0,r_1,\ldots,r_n), t_0}) = 0$; and $F(v_{M,t_0}) = \infty$, for $M \neq (0, r_1, \ldots, r_n)$.

Given the relation and the boundary values, we can implement a dynamic programming algorithm to compute the weight of the shortest path from $v_{(0,r_1,\ldots,r_n),t_0}$ to each vertex column by column and, in each column, from top to bottom. The minimum outcome among the total weights to the last-row vertices is thus our expected result. The optimal link-activation schedule can be derived by a simple backtracking on the corresponding shortest path (referred to as the *last row shortest path*). We then have the following theorem. The proof can be found in [25].

Theorem 1: The centralized dynamic programming solution returns the optimal link-activation schedule.

V. DISTRIBUTED IMPLEMENTATION FOR PRACTICAL SOLUTION

The centralized solution can be computed efficiently on desktop PCs and yield optimal results, which provides useful benchmarks and guidelines for system design and performance evaluation. However, to implement and apply it in EleSense, a series of practical issues remain to be addressed. First, the memory and computation power on a sensor node are very limited comparing with a desktop PC, which makes the algorithms designed for the centralized solution can not be directly applied to such highly constrained hardware. Another issue is that the centralized solution needs all elevator movements to be *known a priori*, while this can not be achieved in reality. In this section, we further discuss these issues and provide our distributed implementation towards the practical solution.

A. Accommodating Hardware and Real-Time Constraints

As mentioned earlier, a sensor node may have limited memory and computation power comparing with a desktop PC. For example, StanfordMote and Imote2 nodes, the two kinds to be evaluated for the GNTVT, have $16MHz$ CPU with $256kB$ RAM and $416MHz$ CPU with $32MB$ RAM, respectively. Such hardware constraints make the algorithms designed for the centralized solution take enormous time to finish or fail due to out of memory, and thus not suitable for scheduling with the elevator moving in real-time. To this end, we design a local search algorithm that can be quickly computed and self-improved during the running time.

The core difference of the local search algorithm is that, instead of exploring the vertices on the time-state graph column by column (as the dynamic programming algorithm for the centralized optimal solution), the local search algorithm visits vertices by an order based on an evaluation function that estimates the remaining weight costs required to achieve the last row. In particular, we use the following evaluation function for a vertex $v_{M,t}$ with $M = (m_0, m_1, \ldots, m_n)$:

$$Eval(v_{M,t}) = \min_{t<t' \leq t_{pause}} \big(CC(t') + \max\big(q(t'-t), TC(t')\big)\big),$$

where t_{pause} is the finish time of the latest known elevator movement (since then the elevator is assumed to pause), $CC(t')$ is the communication cost and $TC(t')$ is the time cost for all the remaining packets to be delivered to the base station. Since the elevator may move during the time of $[t, t']$, we define $E\text{-}MAX(t')$ and $E\text{-}MIN(t')$ as the highest and lowest locations that the elevator appears during this period. Fig. 2a shows an illustration. Sensor nodes are categorized into three sets, namely, "Direct", "Above" and "Below". The Direct nodes have chances to directly transmit data to the base station. Thus their weight costs are estimated as the costs by direct transmissions. The Above nodes are those higher than $E\text{-}MAX(t')$ and need others to help relay packets, where the costs are estimated as delivering data to a base station located at $E\text{-}MAX(t')$. Similar estimation is done for the Below nodes, except that it is computed by the location at $E\text{-}MIN(t')$. The communication cost $CC(t')$ is thus computed as

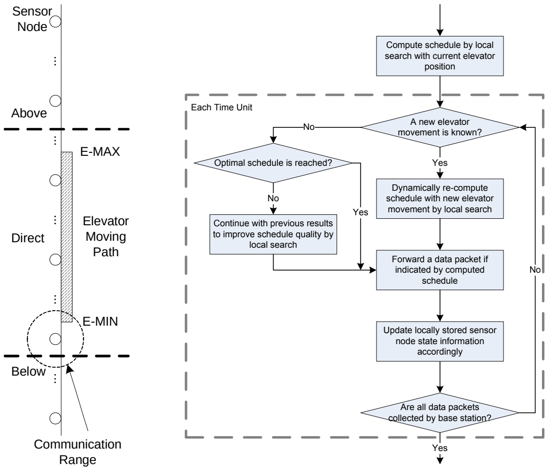
$$\begin{aligned} CC(t') \;=\; & p\cdot\sum_{i=1}^{n} m_i\cdot\Big(I_{[s_i\in Direct]} + \\ & HOP(s_i, E\text{-}MAX(t'))\cdot I_{[s_i\in Above]} + \\ & HOP(s_i, E\text{-}MIN(t'))\cdot I_{[s_i\in Below]}\Big), \end{aligned}$$

where $HOP(a, b)$ is the minimum hop count from a to b. The computation of $TC(t')$ is done by a similar way except that as the data collection may finish before t', we use the maximum value between $TC(t')$ and $q(t'-t)$ as the estimated time cost. We omit further details here due to the space limitation. A full description of the evaluation function design can be found in [25].

As the vertices on a last row shortest path often have better evaluation function values than other neighboring vertices, visiting vertices by an order based on the evaluation function values allows the local search algorithm to find the last row shortest path more efficiently. In addition, according to the computation power and the affordable memory size of a sensor node, we also limits the number of vertices that the local search algorithm can visit so as to finish the computation before the scheduling deadline, i.e., within one time unit. And if a local search can not find a path to the last row before the scheduling deadline (due to a very small search range caused by stringent hardware constraints), it can still choose the path with the best evaluation function value among the explored paths and use it to generate the link-activation schedule for the coming time unit. Moreover, if the optimal schedule is not found in the previous time unit, another local search can be issued at the beginning of the new time unit to continue on with the previous results and further improve the quality until an optimal schedule is found.

B. Scheduling without Apriori Information

Another issue for the centralized solution is that it needs all elevator movements to be known *a priori*. As elevator movements are slower than link-activation scheduling and data transmissions (both of which can be finished within one time unit), in the distributed implementation, each node can dynamically compute the schedule for the next time unit with the current elevator position.



(a) (b)

Fig. 2: An illustration of the practical solution: (a) Evaluation function design; (b) Operation flow of the practical solution at a sensor node. The operations in the dashed rectangle are to be performed within each time unit.

Yet an interesting observation is that, when a floor button on the elevator panel is pressed, the following elevator movements can be precisely known until the elevator reaches the floor indicated by the pressed button. More specifically, in practice the elevator often works in a cycle as follows: It first stays at some location until being recalled by passengers. Then it moves to load the passengers and when the passengers get into the elevator, they would press the floor buttons on the elevator panel to indicate which floors they want to go. After that, the elevator would move to each of those indicated floors to serve the passengers. And when all passengers are served, it will stay at a location until being recalled again. Surely during the elevator serves some passengers, it would also be recalled by others. Although this would further complicate the movements, all the elevator movements however are controlled by the dispatching algorithm preprogrammed in the elevator system. This means that by emulating the elevator dispatching algorithm, we can easily know the elevator movements in the "short future", which enables us to further improve the performance of the distributed implementation.

Therefore, in the practical solution, all nodes initially compute the same link-activation schedule by the elevator position at that time unit. Then following the computed schedule, the nodes forward their data packets and update their local state information accordingly. When a new elevator movement is known, a new schedule is dynamically computed and used to forward packets until all data are collected at the base station.

Fig. 2b summarizes the main operations conducted by each node during a data collection round in the distributed practical solution. In the following sections, we will show that EleSense can achieve excellent performance with very low costs.

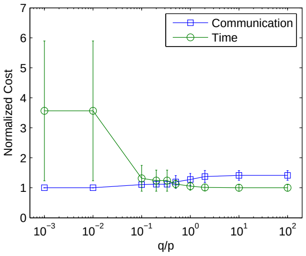


Fig. 3: Impact of ratio q/p on communication and time costs.

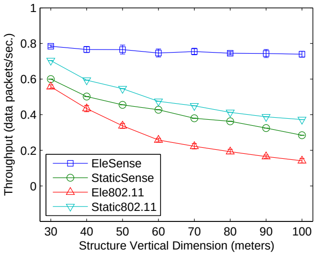


Fig. 4: Overall throughput as a function of structure vertical dimension.

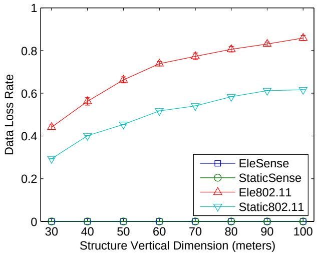


Fig. 5: Data loss rate as a function of structure vertical dimension.

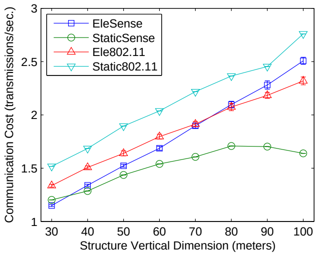


Fig. 6: Communication cost as a function of structure vertical dimension.

VI. PERFORMANCE EVALUATION

We evaluate EleSense by both ns-2 simulations and a case study with real experiments and data sets on the GNTVT. We present the simulation results in this section, and discuss the results of the case study in next section.

A. Methodology

We adopt a typical configuration as follows: The length of a time unit is $1sec$. The queuing buffer size is set to 2. The distance between two neighboring floors is $5m$. On each floor, the representative node is placed at $1m$ to the elevator door with the communication range set to $10m$. We use a random way point model to emulate elevator movements. Specifically, the elevator chooses a destined floor with some probability, moves to it, then pauses for a random period and chooses the next destination. We conduct simulations with different probability distributions, elevator move speeds as well as other setting values, and find the results generally follow the similar trends. Due to the space limitation, we present the results with the destined floor following a uniform distribution and an elevator move speed of $3m/s$. At the initial stage of each simulation, the communication quality of each possible link is explored and those good ones are selected[3]. The IM is then obtained by measuring the simultaneous communication quality through each pair of selected links.

For comparison, we implement three other approaches, namely *StaticSense*, *Ele802.11* and *Static802.11*. StaticSense uses the same cross-layer optimization as EleSense, but the base station is fixed statically at the bottom of the structure. Ele802.11 exploits elevators like EleSense, but using a plain 802.11 MAC layer without the cross-layer optimization. Static802.11 uses a plain 802.11 MAC layer like Ele802.11 and fixes the base station like StaticSense.

Besides the time and communication costs, we are also interested in the following three metrics: 1) *Throughput* is the average number of data packets received by the base station per time unit; 2) *Data loss rate* is the number of data packets that fail to arrive at the base station divided by the number of data packets sent from sensor nodes; 3) *Fairness* is quantified using the classic Jain's fairness index [15], which is defined as $(\sum x_i)^2/(n \cdot \sum x_i^2)$ for demands $x_1, x_2, \ldots, x_n$. For EleSense and Ele802.11, we run 10 simulations on each setting to alleviate the random effects caused by elevator movements. Each data point in the figures thus represents the average of 10 runs with an error bar showing the standard deviation.

B. Impact of the q/p Ratio

To mitigate other uncertainties, we first run the centralized optimal solution to investigate the impact of the q/p ratio. The practical solution will be evaluated in following subsections.

Fig. 3 shows how EleSense performs with different q/p ratios. For ease of comparison, the results are normalized by the corresponding minimum values. When q/p is small ($\leq 10^{-2}$), EleSense yields minimum communication costs but at the expenses of excessive latencies. On the other hand, when q/p grows large (≥ 10), the resulting time costs are minimized while introducing the highest communication costs. Moreover, within the region of $[0.2, 0.5]$, both communication and time costs stay low and keep relatively stable. We thus pick up the middle value $q/p \approx 0.3$ (note x-axis is in log scale) as the default for the remaining evaluations.

C. Scalability with Structure Vertical Dimension

With the default parameter setting, we then conduct simulations on the practical solution to see how EleSense performs with the structure vertical dimension. We let each simulation run for $1000sec$ so as to explore the long-term behavior. In particular, we start a new data collection just after the previous round finishes and stop the simulation at the end of $1000sec$.

Fig. 4 shows the throughput of the four approaches (recall that a time unit is the minimum time for a link to reliably transmit a data packet. Thus ideally the maximum throughput is at most 1 data packet per second). Intuitively, with the vertical dimension increasing, the throughput would slightly drops due to the expansion of the network diameter and/or the increasing possibility of wireless interferences and collisions (as more data packets need to be collected). This is exactly the case for StaticSense, Ele802.11 and Static802.11. EleSense however successfully exploits the elevator and greatly reduces the average distances from the sensor nodes to the base station. Its cross-layer optimization also helps resolve possible interferences and collisions, which makes the throughput almost unchanged with the vertical dimension and achieve the gains of 30.7% to 159.6% over StaticSense and 40.9% to 423.2% over Ele802.11. One interesting observation is that Ele802.11

---

[3] A link is considered good if it can reliably transmit a data packet within one time unit.

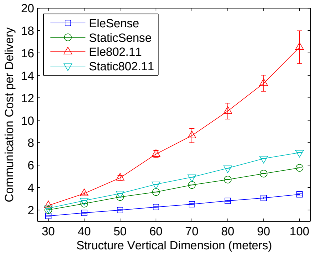


Fig. 7: Communication cost per delivery as a function of structure vertical dimension.

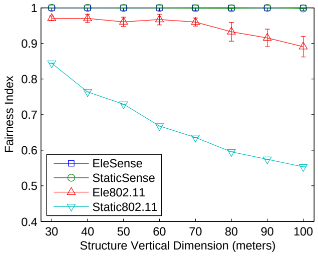


Fig. 8: Fairness index as a function of structure vertical dimension.

performs generally worse than Static802.11. A close look reveals that when the base station moves with the elevator, data traffics may accumulate more quickly by coming from both directions (above and below the base station). This expedites the saturation of the wireless medium near the base station and introduces more significant interferences and collisions, resulting in fewer packets being received at the base station and thus lowering the throughput.

Fig. 5 presents the results on data loss rate. With the cross-layer optimization, both EleSense and StaticSense achieve zero data loss with the built-in retransmission mechanism of the MAC layer, since most interferences and collisions are resolved by the link-activation schedule. On the other hand, the loss rates of Ele802.11 and Static802.11 are generally high and increase with the vertical dimension. This is because larger vertical dimensions will raise the average distances to the base station and thus the possibility of interferences and collisions.

Fig. 6 gives the results on communication cost. It is not surprising that the communication costs rise with the vertical dimension, due to the increasing average distances to the base station. One exception is StaticSense, where the cost first rises and then decreases after the vertical dimension exceeds $80m$. This is because as the vertical dimension increases, more and more data packets will accumulate as relayed to the base station, making it a bottleneck for data collection. At the same time, the cross-layer optimization used in StaticSense successfully suppresses unnecessary transmissions that may cause interferences and collisions, and thus make the overall communication cost drop a little bit. On the other hand, this bottleneck is well handled in EleSense by attaching the base station to an elevator, allowing more transmissions being scheduled for more packet deliveries. This also explains why the cost of EleSense becomes even higher than Ele802.11 when the vertical dimension goes beyond $70m$. As the communication costs of different approaches are afforded for different throughput, we further normalize the communication cost by the throughput to achieve a more fair comparison. The results are shown in Fig. 7. In this figure, although the costs still rise with the vertical dimension, to successfully deliver a data packet, EleSense uses much lower costs (about 58.9% to 73.1% of the runner-up) than the other three approaches.

We also compare the fairness achieved by the four approaches, as shown in Fig. 8. EleSense and StaticSense keep achieving 1.0 in spite of the change of vertical dimension. Ele802.11 also has a fairness index above 0.85, but the fairness index of Static802.11 is generally below 0.85 and drops as the vertical dimension increases. This is because with the presence of wireless interferences and collisions, the packet delivery ratio of a plain 802.11 decreases as the number of traversed links increases, which favors the packets originated from the nodes close to the base station and reduces the fairness.

Elevator Movement	Start Height	End Height	Packets from Each Node	Delivery Ratio
From bottom to top	$0m$	$60m$	20	100%
	$60m$	$120m$	20	100%
	$120m$	$180m$	20	100%
	$180m$	$240m$	20	100%
From top to bottom	$240m$	$180m$	20	100%
	$180m$	$120m$	20	100%
	$120m$	$60m$	20	100%
	$60m$	$0m$	20	100%

TABLE I: Verification Experiments on the GNTVT.

VII. A CASE STUDY WITH THE GNTVT

In this section, we conduct a case study with the GNTVT. As shown in Fig. 1, the tower has an irregular geometry shape with a pathologically extensive vertical dimension. With its floors unevenly spaced, the resulting wireless links as well as interferences and collisions are distributed at great variances. This further varies the capabilities to transmit data to the base station from the nodes at different floors and brings more challenges. To this end, we conduct experiments on the GNTVT to verify the feasibility of EleSense and further evaluate our solutions with the collected data sets.

A. System Deployment and Verification

We adopt the StanfordMote as the data collecting unit and use XStream-PKG $2.4GHz$ RS-232/485 RF modem together with a laptop as the base station. We placed sensor nodes on some floors of the tower and the base station in the No. 2 elevator. Our sensor nodes were equipped with $7dBi$ Buffalo high-gain antenna to enhance the signal strength in the poor construction conditions. The speed of the elevator was around $3m/s$. We adopt high precision accelerometers Tokyo Sokushin AS-2000 for acceleration monitoring. To increase the signal strength and the SNR, a signal amplification and conditioning board is developed using TI PGA202. This board works as an amplification middle-ware between the accelerometer and analogue input of our data collection motes. The sampling rate of the 16-bit Analog to Digital Converter (ADC) on the wireless sensing unit is set to $50Hz$, thus $50 \times 60 \times 2 = 6000$ bytes are generated every minute. These data are first buffered by the SRAM on the wireless unit. Then we divide each $6kB$ data (corresponding to one minute's collection) into 20 packets and transmit them back to the moving base station. It is worth noting that during the experiments, the GNTVT was still in construction. To avoid disturbing the constructions, we were only allowed to access the section below $240m$, which was further divided into 4

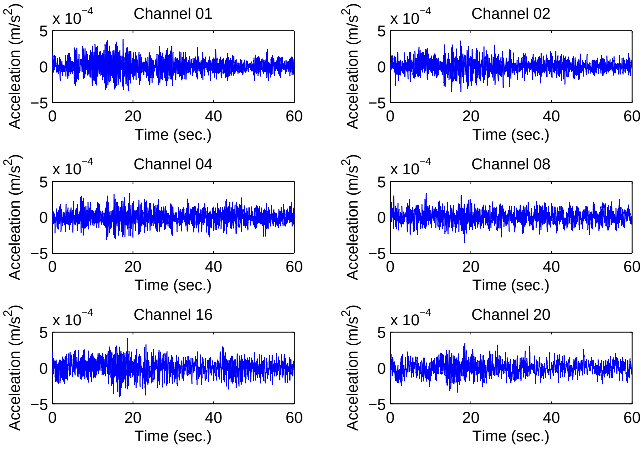


Fig. 9: An illustration of the acceleration data collected by experiments on the GNTVT. Each channel corresponds to the readings from one accelerometer sensor.

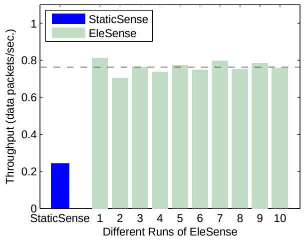


Fig. 10: Overall throughput with real data sets on the GNTVT.

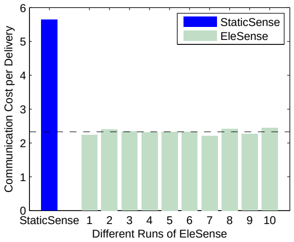


Fig. 11: Communication cost per delivery with real data sets on the GNTVT.

subsections with each containing a part of $60m$. To fully understand wireless communication capabilities, we conduct experiments at both upward and downward directions in each subsection as summarized in Tab. I. From the table, it is clear to see that at each subsection, the base station can successfully receive all the collected data while moving with the elevator at both directions. In addition, we also observed that the wireless transmission could easily reach $55kbps$. All these validate the feasibility of our EleSense framework.

B. Further Evaluation

As the time and region that we can access for conducting experiments were very limited due to the under construction status of the tower, to further evaluate the performance of our solutions on the entire tower, we also conduct emulations with the real data and settings collected on the GNTVT. Our emulations focus on the comparison of EleSense and StaticSense. All the settings are same as the experiments on the tower except that the elevator now moves across the entire tower. For current emulations, we only consider the acceleration data collected by experiments on the GNTVT. Fig. 9 shows an illustration of the collected acceleration data, where each channel corresponds to the readings from one accelerometer sensor.

Similar to the results observed in the simulations, both EleSense and StaticSense schemes achieve zero data loss rate and the fairness index of 1.0. Fig. 10 and Fig. 11 further show the results on throughput and communication cost, respectively, with the dashed line indicating the average of the 10 runs on EleSense. For throughput, EleSense greatly outperforms StaticSense with the average gain as high as 212.7%. Comparing with the simulation results, we find that even under such a stringent scenario, the throughput of EleSense still keeps relatively stable, which further verifies its excellent scalability. For communication cost per delivery, EleSense stays much lower than StaticSense with the average reduction as 58.7%, which also well matches the trend shown in the simulations and demonstrates the superiority of EleSense.

VIII. RELATED WORK

Wired sensor systems have long been used for structural health monitoring; a typical example is the monitoring system of the Ting Kau Bridge [12], Hong Kong. Due to the unique advantages of wireless sensor systems (e.g., low-cost, flexible, robust and readily deployable), a number of recent works have explored the possibility of wireless sensor networks for SHM applications [29][11][4][8][24][3][13][19][26][31]. Among them, prototype WSN systems were developed in [29][11][19]; and the results have generally demonstrated the feasibility to use WSNs in SHM applications. Later, BriMon [4] was proposed for monitoring railway bridges. And a WSN system for monitoring heritage buildings was presented in [3]. For specific problems, a distributed Damage Location Assurance Criterion algorithm (DLAC) was integrated with WSNs in [8] to conduct structural damage localization. Two studies [24][13] focused on optimizing the sensor placement issues by considering specifics of data traffic patterns as well as civil engineering requirements. The monitoring system used by the GNTVT has incorporated some latest advances in the field, and wireless systems are partially adopted. The extensive vertical dimension of the tower, however, poses unique challenges for a full-range wireless sensor systems, especially on efficient data collection. Two preliminary solutions have been investigated in [31] and our previous workshop paper [26]. In this paper, we propose a novel elevator-assisted data collection framework with cross-layer optimization, and provide a fundamental understanding as a step before systematic deployment.

Using controllable mobile base stations to improve the network performance has been studied in different scenarios [10][16][27][23][22] and a theoretical analysis was developed in [20]. In these works, the emphasis is to optimize movement schemes for the base station. In our scenario, the elevator is used to carry passengers and the movements are not controlled by the base station. This makes our scenario unique and calls for an entirely new set of solutions.

There are other related works in the general context of data collection WSNs [1][21][2][7][6][17]. In [1], MAC layer was designed to alleviate the problem caused by traffic accumulation near the base station. In [7] and [6], solutions were explored from the aspects of congestion/rate control and fairness issues. Optimal schemes for data collection with minimum delay were proposed in [21], and later from the theoretical aspect, the authors of [17] investigated the delay and energy tradeoffs for efficient data collection. In [2], a protocol was proposed to achieve ultra low power data gathering, where the MAC layer, topology control and routing are carefully designed to coordinately minimize the energy consumption of the communication subsystem. Recently, cross-layer design has been proposed to improve the performance of wireless networks [5][32][30]. We focus on a specific scenario that applies elevators to assist sensor data collection, which is different from these previous studies.

IX. CONCLUSION AND FUTURE WORK

In this paper, we presented EleSense, a novel framework for high-rise structure monitoring that exploits elevators to reduce the overheads of traffic relaying and balance loads among sensor nodes. As one of the very first studies in this direction, we focused on the most fundamental practical and theoretical constraints. We abstracted the high-rise structure monitoring problem and formulated it as a cross-layer optimization problem with the considerations of link scheduling, packet routing and end-to-end delivery. We presented a centralized optimal solution through dynamic programming. Our design also motivated a distributed implementation to accommodate the hardware capability of a sensor node as well as other practical issues. We evaluated EleSense by ns-2 simulations and a case study with real experiments and data sets on the GNTVT. Both the results demonstrated the superior performance of EleSense.

We are currently conducting more experiments to further evaluate EleSense with the real deployment that is still ongoing on the GNTVT. We believe that many unaddressed factors can be further explored under our EleSense framework. For example, during the work-time, the elevator movement patterns can be learned to further improve the performance. And when the elevator is off-duty (e.g. at night), we can use it as a controllable mobile base station to make the system more efficient. Besides, we also expect to incorporate EleSense into a general data collection protocol.

REFERENCES

[1] G.-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo, "Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks," in *ACM SenSys*, 2006.

[2] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: Ultra-Low Power Data Gathering in Sensor Networks," in *ACM/IEEE IPSN*, 2007.

[3] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment," in *ACM/IEEE IPSN*, 2009.

[4] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, "BriMon: A Sensor Network System for Railway Bridge Monitoring," in *ACM MobiSys*, 2008.

[5] L. Chen, S. Low, M. Chiang, and J. Doyle, "Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," in *IEEE INFOCOM*, 2006.

[6] S. Chen and Z. Zhang, "Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks," in *ACM MobiCom*, 2006.

[7] C. T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," in *ACM SenSys*, 2004.

[8] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke, "A Holistic Approach to Decentralized Structural Damage Localization Using Wireless Sensor Networks," in *IEEE RTSS*, 2008.

[9] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement," in *IEEE SECON*, 2005.

[10] D. Jea, A. Somasundara, and M. Srivastava, "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks," in *DCOSS*, 2005.

[11] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks," in *ACM/IEEE IPSN*, 2007.

[12] J. Ko, Y. Ni, H. Zhou, J. Wang, and X. Zhou, "Investigation Concerning Structural Health Monitoring of an Instrumented Cable-Stayed Bridge," *Structure and Infrastructure Engineering*, 2008.

[13] B. Li, D. Wang, F. Wang, and Y. Ni, "High Quality Sensor Placement for SHM Systems: Refocusing on Application Demands," in *IEEE INFOCOM*, 2010.

[14] Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," in *IEEE INFOCOM*, 2004.

[15] Y. Li, L. Qiu, Y. Zhang, R. Mahajan, and E. Rozner, "Predictable Performance Optimization for Wireless Networks," in *ACM SIGCOMM*, 2008.

[16] J. Luo and J.-P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," in *IEEE INFOCOM*, 2005.

[17] C. Ren, X. Mao, X. Li, P. Xu, and G. Dai, "Efficient Data Collection for Wireless Networks: Delay and Energy Tradeoffs," in *IEEE GlobeCom*, 2009.

[18] T. Schmid, Z. Charbiwala, Z. Anagnostopoulou, M. B. Srivastava, and P. Dutta, "A Case Against Routing-Integrated Time Synchronization," in *ACM SenSys*, 2010.

[19] R. Severino, R. Gomes, M. Alves, P. Sousa, E. Tovar, L. Ramos, R. Aguilar, and P. Lourenco, "A Wireless Sensor Network Platform for Structural Health Monitoring: enabling accurate and synchronized measurements through COTS+custom-based design," in *IFAC International Conference on Management and Control of Production and Logistics*, 2010.

[20] Y. Shi and T. Hou, "Theoretical Results on Base Station Movement Problem for Sensor Networks," in *IEEE INFOCOM*, 2008.

[21] W.-Z. Song, F. Yuan, and R. LaHusen, "Time-Optimum Packet Scheduling for Many-to-One Routing in Wireless Sensor Networks," in *IEEE MASS*, 2006.

[22] S. Tang, J. Yuan, X. Li, Y. Liu, G. Chen, M. Gu, J. Zhao, and G. Dai, "DAWN: Energy Efficient Data Aggregation in WSN with Mobile Sinks," in *IEEE International Workshop on Quality of Service*, 2010.

[23] O. Tekdas, J. H. Lim, A. Terzis, and V. Isler, "Using Mobile Robots to Harvest Data from Sensor Fields," *IEEE Wireless Communications*, vol. 16, no. 1, pp. 22–28, Feb 2009.

[24] F. Wang, D. Wang, and J. Liu, "Traffic-Aware Relay Node Deployment for Data Collection in Wireless Sensor Networks," in *IEEE SECON*, 2009.

[25] ——, "EleSense: High-Rise Structure Monitoring with Wireless Sensor Networks," Simon Fraser University, Tech. Rep., 2010.

[26] ——, "Utilizing Elevator for Wireless Sensor Data Collection in High-Rise Structure Monitoring," in *ACM/IEEE International Workshop on Quality of Service*, 2011.

[27] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station," in *ACM MobiHoc*, 2008.

[28] K. Xu, H. Hassanein, and G. Takahara, "Relay Node Deployment Strategies in Heterogeneous Wireless Sensor Networks: Multiple-Hop Communication Case," in *IEEE SECON*, 2005.

[29] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," in *ACM SenSys*, 2004.

[30] Q. Zhang and Y.-Q. Zhang, "Cross-Layer Design for QoS Support in Multi-hop Wireless Networks," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 64–76, Jan 2008.

[31] T. Zhang, D. Wang, J. Cao, Y. Q. Ni, L. Chen, and D. Chen, "Elevator-Assisted Sensor Data Collection for Structural Health Monitoring," *IEEE Transactions on Mobile Computing*, to appear, 2012.

[32] J. Zhu, S. Chen, B. Bensaou, and K.-L. Hung, "Tradeoff between Lifetime and Rate Allocation in Wireless Sensor Networks: A Cross Layer Approach," in *IEEE INFOCOM*, 2007.