

CitySee: Not Only a Wireless Sensor Network

Yunhao Liu, Xufei Mao, Yuan He, Kebin Liu, Wei Gong, and Jiliang Wang,
School of Software, Tsinghua University

Abstract

CitySee, an environment monitoring system with 1196 sensor nodes and 4 mesh nodes in an urban area, is mainly motivated by the needs of precise carbon emission measurement and real-time surveillance for CO₂ management in cities. Being one of the largest working wireless sensor networks, *CitySee* faces several challenges such as hardware design, software development, platforms, network protocols, and, most important, satisfactory services to users. We share some early lessons learned from this project, illustrate the potential benefits and risks of current solutions, and discuss the possible extensions of *CitySee* applications.

Recently, natural disasters and extremely abnormal climate situations happen frequently and globally, the culprit of which is the exacerbation of global warming. It is a consensus that both governments and individuals should take action to control greenhouse gas emission, especially emission of CO₂, which is mainly caused by human activities. According to the statistical data, arguably more than 80 percent of CO₂ emissions originate in cities, while cities only occupy less than 2.4 percent of the global land mass [1]. Hence, understanding the relationships between the form and pattern of urban development and the carbon cycle is crucial for estimating future trajectories of greenhouse gas concentrations in the atmosphere and facilitating mitigation of climate deterioration. There are generally two strategies admitted by most countries for computing/estimating pollutant (e.g., CO₂) emissions of factories, areas, or countries. The first one is to estimate CO₂ emissions based on raw material (e.g., fossil oil and coal) consumption, which is adopted by the European Environment Agency when it ranks countries by their CO₂ emissions every year. The other one, called inference-based carbon sensing technology, is utilized to measure pollutant gas fluxes [2], like CO₂ and NO₂. Both estimation-based and inference-based methods are less accurate and do not satisfy the real-time requirements for pollutant gas monitoring. *CitySee* is designed as an alternative approach to directly measure pollutant emissions of large-scale areas accurately and thoroughly in a real-time and long-term manner. It integrates several thousands of sensors with mature technology and small individual wireless nodes to make us capable of interacting with the physical world. In our first phase of implementation, 1200 nodes (1196 sensor nodes and 4 wireless mesh nodes) are deployed in the urban area of Wuxi City, China, so as of today, July 2013, multidimensional environmental data (CO₂, NO₂, temperature, humidity, light level, location, etc.) have been continuously collected in a real-time manner since August 2011. *CitySee* faces many challenges, each of which corresponds to several issues that need to be properly addressed.

- **Hardware design:** To facilitate large-scale and affordable deployment, we need to carefully consider both the archi-

ture design and the encapsulation design [3] for terminal nodes. Since all 1200 nodes are designed to be deployed in outdoor environments without necessary protection, robustness is a big concern.

- **Software design:** *CitySee* contains several levels of software [4], including hardware drivers, embedded software for data collection, data processing, and routing [5], as well as application software for providing rich and friendly services to users. In addition, the software of *CitySee* should shield the difference of heterogeneous operating system and hardware platforms according to compatibility and expansibility.
- **Network design:** To the best of our knowledge, *CitySee* is currently the largest working wireless sensor network (WSN). We have to address a series of key issues so as to keep the entire system running healthily and smoothly, such as *sensor deployment* [3], *data loss* [6], *energy efficiency* [5], *network management and diagnosis* [7].
- **Pervasive services:** The ultimate goal of *CitySee* is to provide pervasive services to both governments and individual users. In other words, besides various of applications supported by all kinds of sensory data, *CitySee* also provides other types of services, like localization service [8] utilizing wireless signals of deployed sensors, or short video information by equipping small cameras into nodes.

In this article, we briefly go through each part of *CitySee* and take a first step toward how to combine WSNs with current leading technologies (e.g., cloud computing) in order to provide satisfactory pervasive services to both network designers and users with respect to scalability, performance, privacy, cost savings, etc.

Related Work

One of the first studies on WSNs for habitat monitoring was reported by Mainwaring *et al.* in [9], in which an instance of WSN for monitoring seabird nesting environments and behaviors was described. The authors mainly investigate hardware design of sensors, network architecture, capabilities for remote data access and management, etc. An early surveillance application using a power-constrained sensor network

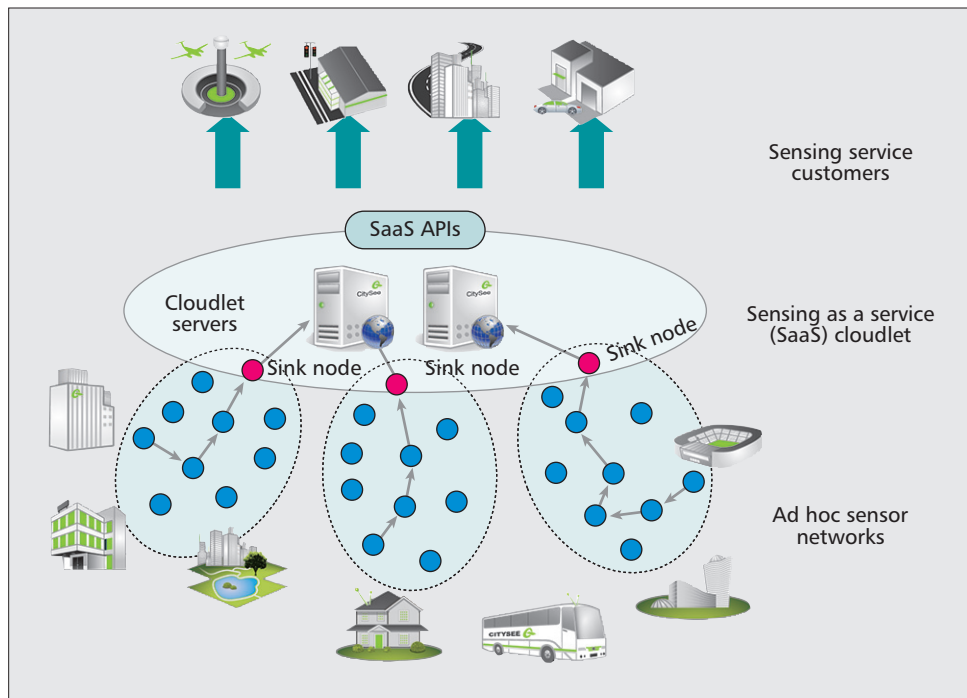


Figure 1. Three-layer service structure of CitySee.

was proposed by Vicaire and He *et al.* in [10]. The authors present the design and implementation of multidimensional power management strategies in VigilNet, and further introduce a novel tripwire service with effective sentry and duty cycle scheduling so as to increase the system lifetime. There are also some other important WSN applications [11, 12] for environmental surveillance.

In our early project *GreenOrbs* [6], we first deploy 330 wireless TelosB nodes equipped with temperature, humidity, and light sensors in the forest, for studying canopy closure of the forest, and later more sensors are employed. Sensor nodes in *GreenOrbs* are intensively dense due to the canopy measuring requirements. Indeed, since there are not many vehicles and very few people in the area, the environment of *GreenOrbs* is not as complicated as in a city.

CitySee can also be viewed as the second phase of *GreenOrbs*, and it has 1200 working nodes now, including 1196 wireless sensor nodes (each has multiple sensors) and 4 wireless mesh nodes. Besides the challenges brought by the larger scale, we focus more on coupling WSNs with other leading technologies (e.g., cloud computing) in order to provide satisfactory services to both network designers and service users.

Solutions

System Infrastructure

CitySee is currently designed as a pervasive service that integrates both the underlying wireless sensor network techniques (“sensing IaaS”; IaaS stands for infrastructure as a service) and the upper-layer cloud computing applications, that is, a sensing as a service (SaaS) cloudlet. Here, the reason for us to use “cloudlet” rather than “cloud” is because our current cloud computing architecture is not a full-fledged cloud facility yet, and it is specially simplified/trimmed for our concentrated sensing services. Meanwhile, we are making continuous efforts to strengthen it into a real cloud, as depicted in Fig. 1.

At the underlying layer, various kinds of sensors deployed across buildings and plazas constitute the sensing IaaS. At the upper layer, the SaaS cloudlet consists of both cloudlet servers and sink nodes, as in *CitySee* sink nodes not only aggregate and

forward raw sensing data (i.e., sink nodes’ conventional functionality in a WSN), but also preprocess the data before synchronizing them to cloudlet servers. Finally, cloudlet servers form a series of SaaS application programming interfaces (APIs) that facilitate the access, use, and programming of sensing service customers.

Hardware

We design and implement two types of nodes, wireless sensor nodes equipped with different types of sensors (abbreviated to sensor nodes) and wireless mesh nodes (abbreviated to mesh nodes), respectively. The hardware platforms of sensor nodes are based on TelosB. A wireless sensor node is equipped with an MSP430F1611 processor and a CC2420 radio chip compliant with the 802.15.4 protocol at 2.4 GHz. Each node has one or multiple sensors, such as temperature, humidity, CO₂, NO₂, wind velocity and direction, according to various application requirements. All nodes are encapsulated with industrial grade design in order to adapt to hostile outdoor environments. Sensor nodes mainly take charge of sensing the environments, packetizing sensing data, and further sending data packets to one of sink nodes (each of which connects to a mesh node directly) through one- or multi-hops (i.e., they form a wireless ad hoc network). Due to the outdoor deployment environments, during the past two years most of the sensor nodes have either had their batteries changed one or more times or have been replaced.

CitySee currently has four mesh nodes constructing a wireless ad hoc network, with one of them selected as the final sink. A mesh node mainly consists of an ARM7 processor and a network card of type WL017MP, using the 802.11a protocol at 5.8 GHz. Considering the difficulty of power supply, a mesh node could be powered by either electricity, net wire, or solar panels.

Software on Nodes

We develop software for different types of sensor nodes on top of TinyOS 2.1.1, which consists of the following major components. First, we implement the link estimation component using the four-bit link estimation method to regularly maintain a neighbor table. Second, we use the default Low

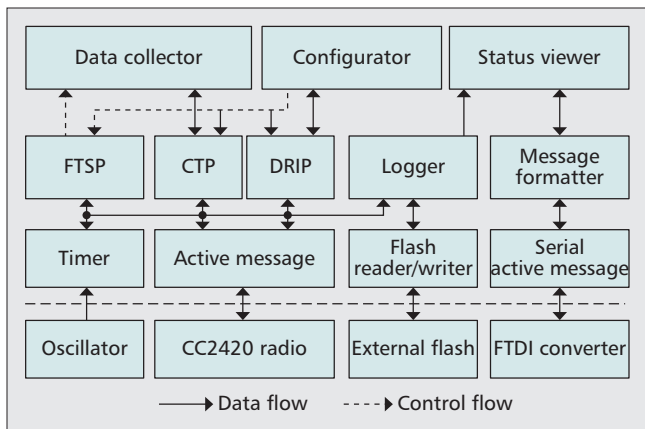


Figure 2. Architecture of embedded software.

Power Listening medium access control (MAC) protocol of TinyOS to reduce the energy consumption. Third, the multi-hop routing component is implemented based on the Collection Tree Protocol (CTP) [13] for data collection. Fourth, we apply the Drip protocol to disseminate key system parameters in terms of the dissemination component, such as transmission power levels, sampling frequencies, and duty cycles. In *CitySee*, a sensor node is programmed to sample the environment every 10 minutes and then issue data packets to a sink. The detailed structure of the software running on a sensor node is shown in Fig. 2.

Connectivity

One of the biggest challenges of *CitySee* is to maintain the connectivity of all 1200 nodes, especially when harsh conditions exist. For *CitySee*, the exact locations for all CO₂ sensor nodes and some given types of environmental sensors have been pre-designated by environmentalists. As the sensing operation of a CO₂ sensor is very energy consuming, we normally do not let the nodes equipped with a CO₂ sensor (100 of them) relay packets for others, while the other type of nodes perform both sensing and relaying operations. In addition, due to the complicated outdoor conditions and physical constraints of urban areas (e.g., buildings and artificial lakes), on one hand, there are some places where we cannot deploy any nodes; on the other hand, two nearby sensor nodes may not be able to communicate with each other due to signal blocking, reflection, and so on.

We formulate this issue as a geometric Group Steiner Tree with Holes problem and propose a 2-approximation algorithm to solve it [3]. Our main idea is to define and construct legal components depending on the first deployment (determined by the environmentalists), based on which a secondary deployment is conducted while minimizing the total number of relay nodes needed as the optimization objective. We apply the proposed strategies to *CitySee* and part of the deployment situation of *CitySee* as shown in Fig. 3.

Data Collection

The fundamental goal of *CitySee* is to monitor and collect environment data frequently, so keeping the integrity and real-time properties of sensory data is necessary and important. In addition, status data of nodes and the network are necessary as well since grasping the running status of both a single node and the network is critical for maintaining the system.

Protocols — In *CitySee*, we mainly employ CTP [13] for multi-hop data collection; that is, each node is planted with an agent, which senses the environment periodically and issues data packets to a sink node through one hop or multiple hops; all sensor nodes act as leaf nodes only. We collect three types of data packets, each of which contains different types of information. A packet with type C_1 has two categories of information:

- Sensing data: temperature, humidity, light or CO₂ concentration values
- Routing information: including path-ETX [13] from the original source of the packet to some sink node, through which we are able to recover the complete routing path of any packet arriving at a sink node

A packet with type C_2 records local information for each node, such as routing table information, including IDs and received signal strength indicator (RSSI) values, from its neighbors, and sends ETX estimation values of links to its neighbors. A packet with type C_3 contains more detailed information on a single wireless node. For instance, the *CPU counter* records the accumulated task execution time, the *radio counter* indicates the accumulated radio-on time, the *transmit counter* depicts the accumulated number of transmitted packets, the *receive counter* describes the accumulated number of received packets, and the *loop counter* tells us the accumulated number of detected loops.

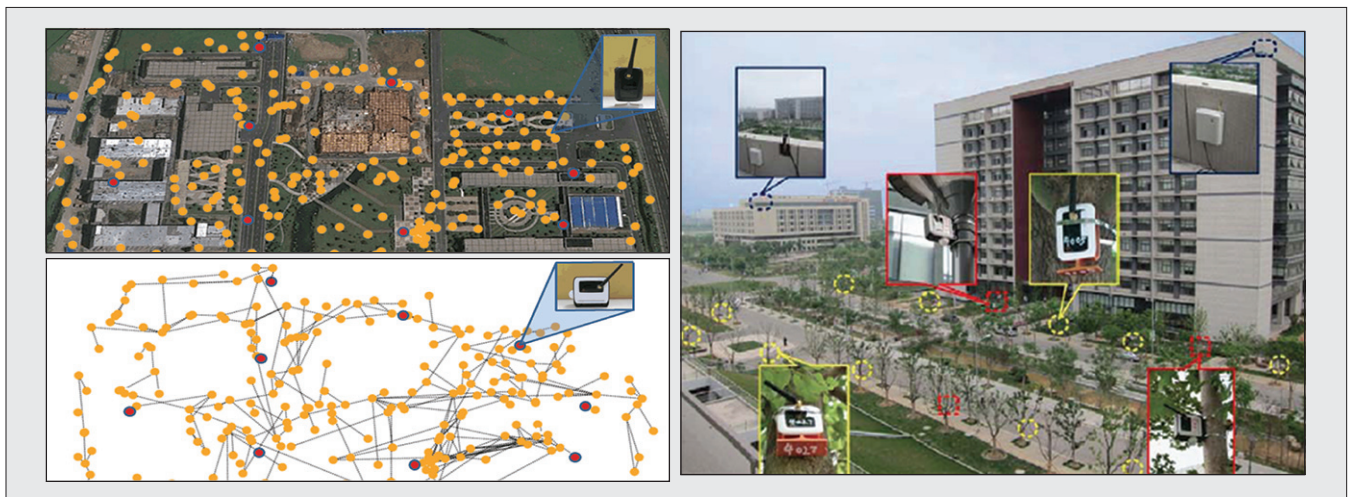


Figure 3. Part of the deployment situation of *CitySee*.

Based on status data, *CitySee* provides network management services to network administrators by showing the real-time running status of both the entire network and an individual node. For instance, the left part of Fig. 4 plots the partial logical topology of 1200 nodes during the last duty cycle; each individual node is shown as a circle whose area indicates the number of packets it has transmitted for the last 10 minutes. After clicking a node, the whole path along which its last packet walks is shown. The right part of Fig. 4 indicates the measurement matrix (with more than 20 indices) of a single node, including the number of tasks it has posted and executed, the number of duplicated packets, and so on.

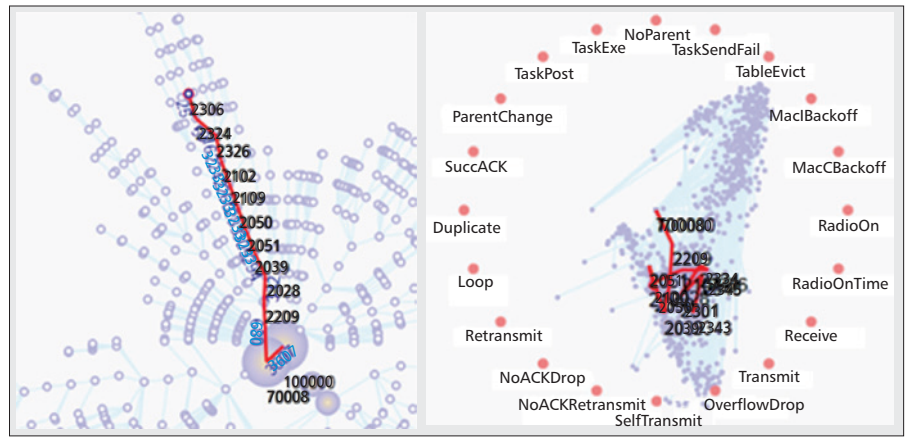


Figure 4. The logical topology of part of 1200 nodes and the multidimensional matrix of the status of a single node.

Potential Benefits of Using Cloud Computing Strategies — At a first glimpse, it seems that data collection in a WSN is kind of easy since every node only needs to send data packets to some node(s) by pre-designed rules. However, it is indeed quite complicated, especially when resources of a wireless node are limited considering two batteries will support more than one year, while wireless links in the network are very unstable, so many retransmissions are necessary. In *CitySee*, the traffic loads of nodes with different roles (e.g., relay nodes and leaf nodes) are quite different, depending on the physical environment and routing protocols. For example, the number of tasks executed by a node close to the sink node could be up to 8742 in 10 minutes, while around 1/4 of wireless nodes have an average number less than 8. Figure 5 shows the traffic load of nodes in *CitySee*, from which we can see that the traffic load of a node is not merely determined by its physical location or assigned roles. In *CitySee*, two types of data collection strategies are designed: active report and passive query. When the active report strategy is adopted, a node chooses one or more nodes as its relay nodes when it has packets to transmit. On the contrary, passive query strategy is activated in some cases. For instance, when a node detects abnormal data packets from nearby nodes, it issues a query to the latter one and requires the node to resend its packets.

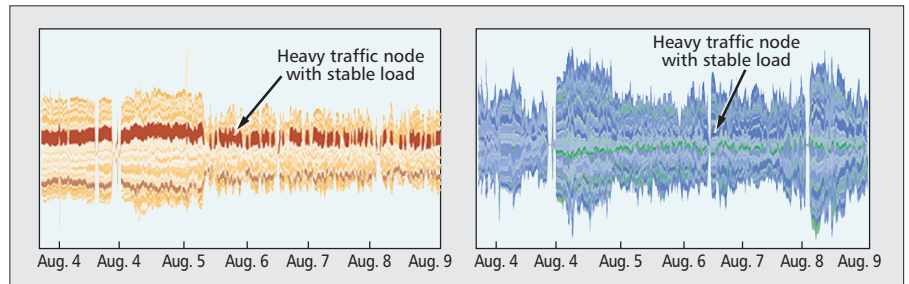


Figure 5. Traffic load of nodes.

After packets arrive at the sink nodes, data synchronization into the cloudlet servers is distinct from conventional data synchronization (e.g., Dropbox, Amazon S3, and Microsoft Azure) in commercial cloud storage services due to the fact that data flows in WSNs are usually frequent, short, and location-sensitive. In the presence of such data flows, coarse-grained cloud synchronization mechanism design may lead to severe traffic overuse. For example, even one byte's sensing data synchronization (from a sink node to a cloudlet server

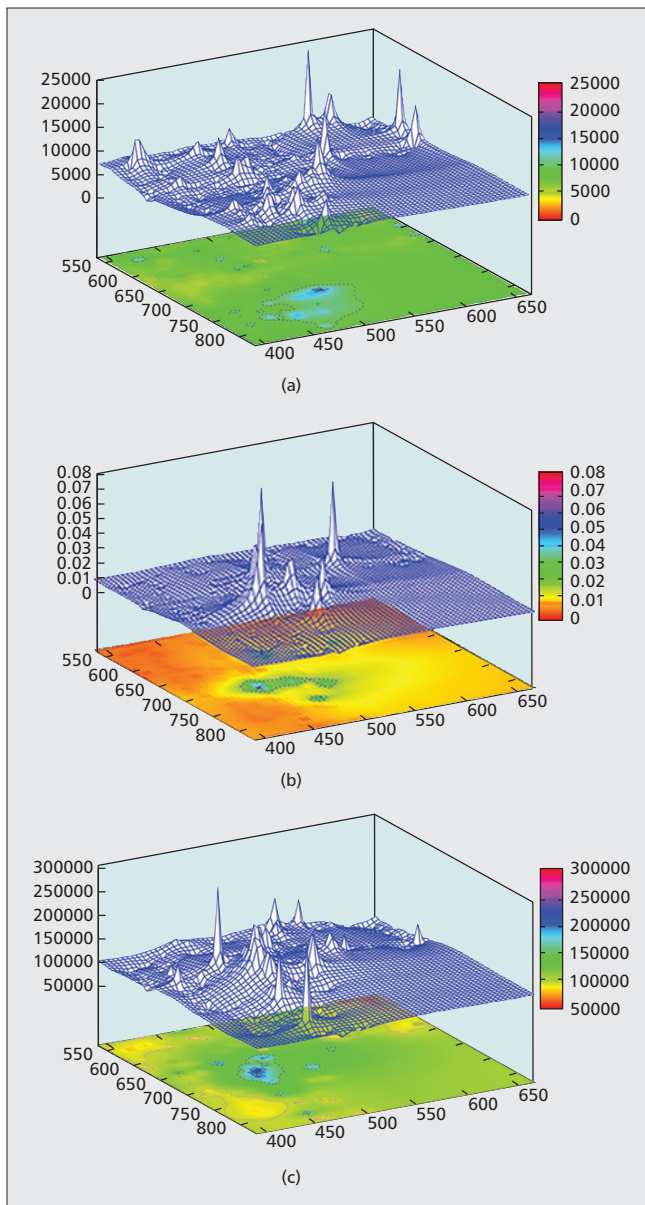


Figure 6. Data for network diagnosis and management: a) number of parent changes; b) radio duty cycle; c) number of packets transmitted.

wait around 15 s, which is a terrible user experience without a doubt. In addition, a single database cannot satisfy the reliability requirements for *CitySee* since the power shutdown or physical damage to hard disks leads to temporary or even unrecoverable data loss. Clearly, utilizing the cloud can alleviate this situation. Here, we face two choices, a private cloud (i.e., we build our own cloud using a number of high-performance servers) or the public cloud (e.g., obtaining services from Amazon or Google directly). The question is that the advantages and disadvantages of both private clouds and public clouds are obvious. For example, to design our own private cloud, it seems that we only need to pay more for hardware instead of paying service fees to cloud service providers. However, it is hard to achieve as good performance as the cloud service providers do in terms of efficiency, reliability, and scalability since the latter have relatively mature technologies. On the contrary, if we totally trust the public cloud, besides the high cost, do we really trust the security of their services, especially for sensitive data?

After counting the cost, our next step for *CitySee* is to combine both a private cloud and the public cloud in order to both obtain and provide the best cloud services, that is, building our own private cloud and using the public cloud at the same time.

Network Diagnosis and Management

CitySee has a long-term running objective, and any physical modification of the network (e.g., replacing individual nodes) is pretty costly, so it is critical and necessary to learn the running status of the entire network as well as each individual node. In order to collect key metrics, such as radio duty cycles, and the number of packet transmissions and receptions, and provide visibility to the system, we further design and implement the network management and diagnosis component. We design and implement PAD [15], a probabilistic and passive diagnosis approach for inferring the root causes of abnormal phenomena. PAD employs a packet marking algorithm for efficiently constructing and dynamically maintaining the inference model without incurring additional traffic overhead for collecting desired information. Using PAD, we design a number of indices to evaluate the health of *CitySee* (e.g., data reception ratios, total number of tasks executed, routing loops detected, traffic analysis), which are shown in Fig. 6.

Through continuously analyzing the running status of *CitySee* for the last two years, we have many interesting observations. According to the link level study, we seek to answer several fundamental questions: what are the temporal and spatial characteristics of links, what causes link performance degradation, and what is the impact of link performance on wireless ad hoc network routing? The interesting key findings include:

- The performance of intermediate links is the most unpredictable, and some links exhibit highly periodic patterns.
- The width of the reception “transitional region” is much larger than those reported in previous experiments, indicating that an outdoor environment might have a greater impact on the link performance.
- Differing from previously reported results, link performance degradation has a relatively weak correlation with the corresponding RSSI values fluctuating near the noise floor.
- Although individual links exhibit high dynamics, the proportions of good, intermediate, and poor links are fairly stable over time.
- Low-performance links are often wrongly selected for routing due to the inefficiency of current routing algorithms, rather than performance degradation of the majority of links.

Mobility and Localization

As mentioned earlier, *CitySee* is not simply designed as a wireless sensor network, and we put much effort into its scalability and compatibility. For instance, besides the static deployed sensor nodes, we design mobile sink nodes (handsets based on ARM and TelosB) to conduct mobile data collection. In addition, by introducing the concept of node localization and analyzing the conditions for a node being uniquely positioned under a certain sensor network topology, we utilize the infrastructure of sensor nodes and mesh nodes of *CitySee* to provide localization services to mobile users as well. Furthermore, we are combining one of our ongoing projects *LiFS* [8], a localization system based on off-the-shelf WiFi infrastructure and mobile phones, with *CitySee* in order to provide more accurate localization services to users. Our main idea is to use all deployed nodes to provide the calibration of fingerprints in a crowdsourcing and automatic manner, as we did in *LiFS*. Con-

sidering that the energy issue of WSNs is nontrivial, we enable both sensor nodes and mesh nodes to be powered by solar energy directly. With energy-harvesting technology, we are able to dig up more about the routing and other protocols in WSNs, which enhances the Sensing IaaS ability of *CitySee*.

Conclusions

We present *CitySee*, an environment-monitoring system using a large-scale WSN in an urban area in Wuxi, China. We briefly go through the solutions of *CitySee* with respect to hardware design, software design, network management and diagnosis, and pervasive services.

With the increment of mobile users, mobile crowdsourcing has received much attention. Since both sensor deployment and network maintenance often incur unacceptable cost, we believe that to partially use the existing devices in people's hands under the concept of crowdsourcing and participatory sensing is going to be one of the further directions. Furthermore, as WSN and Internet of Things techniques become more mature, sensing is expected to be industry-oriented and programmer-friendly (not just for domain professionals). Therefore, we have been enhancing and transforming the *CitySee* project into a service (i.e., sensing as a service) that facilitates its users/customers. In other words, *CitySee* is evolving toward an innovative, pervasive, and easy-to-use cloud service platform.

Acknowledgment

This work is supported by the NSFC Major Program 61190110, the NSFC Distinguished Young Scholars Program under grant 61125202, NSFC under grant 61272426, China Postdoctoral Science Foundation funded project under grant 2012M510029 and 2013T60119, National High-Tech R&D Program of China (863) under grant 2011AA010100, and China 973 Program under grant 2011CB302705.

References

- [1] G. Churkina, "Modeling the Carbon Cycle Of Urban Systems," *Ecological Modelling*, vol. 216, no. 2, 2008, 107–13.
- [2] A. Chédin *et al.*, "The Feasibility of Monitoring CO₂ from High-Resolution Infrared Sounders," *J. Geophysical Research: Atmospheres*, vol. 108, no. D2, 2003, pp. 1984–2012.
- [3] X. Mao *et al.*, "Citysee: Urban CO₂ Monitoring with Sensors," *Proc. 31th IEEE Int'l. Conf. Comp. Commun.*, 2012, pp. 1611–19.
- [4] W. Gong *et al.*, "Quality of Interaction for Sensor Network Energy-Efficient Management," *Computer J.*, 2013.
- [5] J. Wang *et al.*, "Qof: Towards Comprehensive Path Quality Measurement in Wireless Sensor Networks," *Proc. 30th IEEE Int'l. Conf. Comp. Commun.*, 2011, pp. 775–83.
- [6] Y. Liu *et al.*, "Does Wireless Sensor Network Scale? A Measurement Study on Greenorbs," *Proc. 30th IEEE Int'l. Conf. Comp. Commun.*, 2011, pp. 873–81.
- [7] Y. Liu, K. Liu, and M. Li, "Passive Diagnosis for Wireless Sensor Networks," *IEEE/ACM Trans. Net.*, 2010, vol. 18, pp. 1132–44.
- [8] Z. Yang, C. Wu, and Y. Liu, "Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention," *Proc. 18th ACM Annual Int'l. Conf. Mobile Computing and Networking*, 2012, pp. 269–80.
- [9] A. Mainwaring *et al.*, "Wireless Sensor Networks For Habitat Monitoring," *Proc. 1st ACM Int'l. Wksp. Wireless Sensor Networks and Applications*, 2002, pp. 88–97.
- [10] P. Vicaire *et al.*, "Achieving Long-Term Surveillance in Vigilnet," *ACM Trans. Sensor Networks*, vol. 5, no. 1, 2009.
- [11] L. Selavo *et al.*, "Luster: Wireless Sensor Network for Environmental Research," *Proc. 5th ACM Int'l. Conf. Embedded Networked Sensor Systems*, 2007, pp. 103–16.
- [12] G. Tolle *et al.*, "A Macroscopic in the Redwoods," *Proc. 3rd ACM Int'l. Conf. Embedded Networked Sensor Systems*, 2005, pp. 51–63.
- [13] O. Gnawali *et al.*, "Collection Tree Protocol," *Proc. 7th ACM Conf. Embedded Networked Sensor Systems*, 2009, pp. 1–14.
- [14] Z. Li, Z. Zhang, and Y. Dai, "Coarse-Grained Cloud Synchronization Mechanism Design May Lead to Severe Traffic Overuse," *IEEE Tsinghua Science and Technology*, 2013.
- [15] X. Miao *et al.*, "Agnostic Diagnosis: Discovering Silent Failures in Wireless Sensor Networks," *Proc. 30th IEEE Int'l. Conf. Comp. Commun.*, 2011, pp. 1548–56.

Biographies

YUNHAO LIU [SM'06] received his B.S. degree in automation from Tsinghua University, Beijing, China, in 1995, and M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, in 2003 and 2004, respectively. He is a professor with the School of Software, Tsinghua National Lab for Information Science and Technology, and director of the MOE Key Lab for Information Security, Tsinghua University.

XUFEI MAO [M'10] (xufei.mao@gmail.com) received his Ph.D. degree in Computer Science from Illinois Institute of Technology, Chicago, in 2010. He received his M.S. degree (2003) in computer science and Bachelor's degree (1999) in computer science from Northeastern University and Shenyang University of Technology, respectively. He is currently with the School of Software, Tsinghua University. His research interests span wireless ad hoc networks, and pervasive computing.

YUAN HE received his B.E. degree from the University of Science and Technology of China, his M.E. degree from the Institute of Software, Chinese Academy of Sciences, and his Ph.D. degree from Hong Kong University of Science and Technology. His research interests include sensor networks, peer-to-peer computing, and pervasive computing.

KEBIN LIU received his B.S. degree from the Department of Computer Science at Tongji University in 2004, and his M.S. and Ph.D. degrees from Shanghai Jiaotong University in 2007 and 2010 respectively. He is currently an assistant researcher in the School of Software at Tsinghua University. His research interests include sensor networks and distributed systems.

WEI GONG received his B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2003, and his M.S. and Ph.D. degrees from the School of Software and Department of Computer Science and Technology at Tsinghua University in 2007 and 2012, respectively. His research interests include wireless sensor networks, RFID applications, and mobile computing.

JILIANG WANG received his Ph.D. degree from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. He received his B.E. degree from the Department of Computer Science from the University of Science and Technology of China. He is currently with the School of Software at Tsinghua University. His research interest includes wireless sensor networks, network measurement, and pervasive computing.