

Networking Strategies for Structural Health Monitoring in Wireless Sensor Networks

Yujin Lim and Jaesung Park*

*Department of Information Media, Department of Information Security
University of Suwon, San 2-2, Wau-ri, Bongdam-eup, Hwaseong,
Gyeonggi-do, 445-743, Korea
{yujin, jaesungpark}@suwon.ac.kr*

Abstract

Structural health monitoring (SHM) systems using wireless sensor networks (WSNs) improve the safety and reliability of civil infrastructures by detecting damages before they reach a critical state. To achieve this, numerous challenges range from software design to hardware design. However, the intrinsic networking requirements for SHM have been overlooked in current SHM design. Therefore, we analyze wireless networking issues to satisfy the requirements of SHM. Then, the state of the art in the networking issues is reviewed including reliable data transfer, time synchronization, and power management. Finally, we discuss opportunities influencing WSN development for SHM in the future.

1. Introduction

Structural health monitoring (SHM) is estimating the state of structural health, or detecting the changes in structure that effect its performance. It can be adopted for newly constructed structures and existing structures to understand structural health condition and prevent disaster by prior detection [1].

A traditional SHM using wired network has three drawbacks in that (1) the high cost of installation and disturbance of the normal operation of the structure due to wires having to run all over the structure, (2) the high cost of equipment, and (3) the high cost of maintenance. In order to overcome the drawbacks of the wired SHM system, the SHM system using wireless sensor networks (WSNs) has emerged. The WSN-based SHM system employs WSNs to transfer data between sensors, between sensor and sink, and between sink and data collecting server, as shown in Figure 1. The WSN-based SHM has three advantages in that (1) the cost of cable installation and maintenance is reduced dramatically, (2) the onboard microprocessor of the wireless sensor can facilitate efficient distributed data processing, and (3) WSN makes the updating, adding, moving, and replacing of sensors easy after the initial installation. The network can be reorganized quickly without disturbing the original data collecting operation.

The WSN-based SHM system (hereafter we call it SHM) is an integration of sensory system, data collecting system, communication system, data processing system and damage detection and modeling system [2]. A typical architecture of the SHM is shown in Figure 1. While the data communication between data collecting server and main control center is based on wired network, the data communication between the sensor nodes/sinks and data collecting server is based on WSN.

* Corresponding author

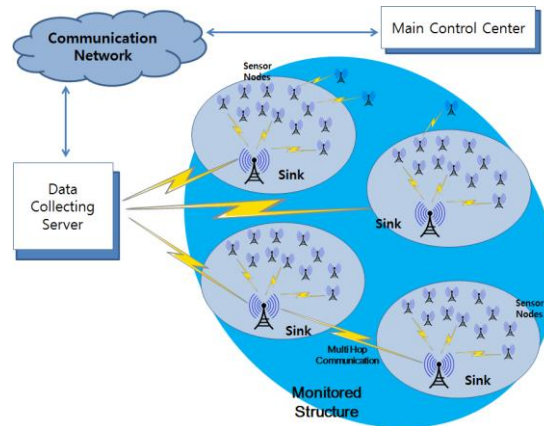


Figure 1. General Architecture of a WSN-based SHM

Typically, WSN enables many monitoring applications with low data rate, small data size, low duty cycle, and extreme emphasis on low power consumption. However, the SHM requires high data rate, large data size, and a relatively high duty cycle. In order to satisfy the requirements in the SHM, WSN should handle numerous challenges before the real deployment. These challenges range from software design to hardware development. We enumerate major challenges in five types [3]. Firstly, hardware packaging is considered to ease installation, to retain high sensibility, to protect components from harsh environmental conditions, and to support capability extension. Secondly, scarce resources, such as energy, memory, and computing time-slots, need to be managed efficiently. Thirdly, although wireless communication has many advantages, it is also subject to several drawbacks. It is difficult to achieve reliable data transmission and time synchronization. Fourthly, in contrast to raw data acquisition policy in wired sensor network, WSN requires in-network information processing to save energy and increase information throughput per bit. Finally, in order to produce generic solutions for low cost and wide deployment, it is important to achieve good interoperability and compatibility between different platforms.

In this paper, we focus on the wireless communication problem and present a survey on technical issues in WSNs for the SHM. Section 2 introduces software architecture of a WSN node for the SHM. In the architecture, we handle three issues – reliable data transfer, time synchronization, and power management in Section 3-5, respectively. Finally, conclusion is drawn in Section 6.

2. Software Architecture of a WSN Node

The WSN is mainly composed of sensor nodes and sink nodes. The sensor node measures structural state and sends the measured data to the sink node. The sink node has more computational power and larger storage than the sensor node. It collects the measured data from sensor nodes. Then, if it is necessary, it processes the collected data through local data filtering, data compression or data aggregation. Finally, it forwards the processed data to the data collecting server. Figure 2 shows the software architecture of a WSN node [4,5]. In the figure, *Best-effort Single-hop Communication* module is used to provide fair wireless medium access. *Broadcast* module is for the reliable command dissemination with low latency. *Routing* module provides a best-effort multi-hop routing. Medium access control algorithm for single-hop communication cannot solve the reliable transfer problem due to the hidden terminal effect. *Reliable Transfer* module provides end-to-end acknowledgement and rate control mechanism to handle the packet loss and prevent the packet loss, respectively. Rate mismatches may cause queues along the path to

overflow, leading to packet loss, wasted energy, and additional end-to-end retransmission. Because a lot of WSN nodes work in environments where a direct power supply is not available, enhancement of a node lifetime by using the low-power technology, *Power Management* module, is one of major issues. In the SHM, there are many requirements regarding synchronous data acquisition of the measured information. *Time Synchronization* module controls jitter to guarantee time synchronization in the node and across the network. In this paper, we address the issues of reliable transfer, time synchronization, and power management.

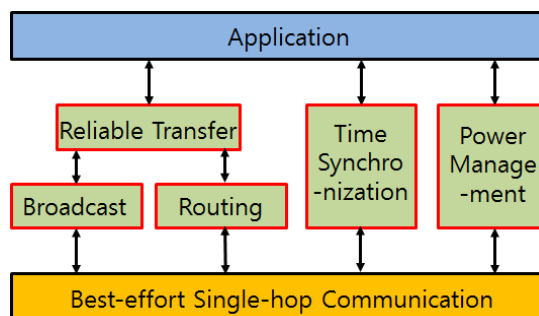


Figure 2. Software Architecture of a WSN Node

3. Reliable Data Transfer

The SHM system needs the data from all the sensors to calculate the entire system response and it is less tolerant to the data loss. Also the data are collected after a long period of time, and loss of data may cause a long delay in generating complete system response which will question the performance of the system. Thus, reliable data transfer is one of major requirements of the SHM system.

Flush [4, 6] is a reliable, high throughput, bulk data transport protocol for the SHM. The system is deployed at Golden Gate Bridge in San Francisco, USA. Flush has two goals: reliable delivery and minimizing transfer time. However, the nature of wireless communication brings several challenges for the goals: lossy links, inter-path interference, intra-path interference, and transient rate mismatches. For the reliable delivery, Flush uses a sink-initiated control protocol to coordinate the transfer with end-to-end selective negative acknowledgements and retransmissions. In other words, a sink keeps track of all received packets. When the source has finished sending data, the sink sends a NAK packet for the lost packets. Then, the source retransmits the lost packets. This process continues until the sink has received every packet. For the minimization of transfer time, the rate control algorithm is proposed to deal with the challenges, such as inter-path interference, intra-path interference, and transient rate mismatches. The inter-path interference occurs when two or more flows interfere with each other. The intra-path interference occurs when transmission of the same packet by successor nodes prevents the reception of the following packet from a predecessor node. The transient rate mismatches may cause queues along the path to overflow and leads to packet loss. The rate control algorithm of Flush has focused on the intra-path interference and transient rate mismatches. In order to solve the intra-path interference, a node should transmit only when its successor is free from interference through the pipelining with the spatial reuse. In order to solve the transient rate mismatches, the sending rate of a node cannot exceed the sending rate of its successor. However, Flush ignores the inter-path interference. Besides, it obtains necessary information by snooping to minimize extra control packets but it is demanding extra power consumption.

In order to cope with the inter-path interference, the interference-aware fair rate control (IFRC) is designed. In IFRC, each node locally detects all flows that can contend for channel capacity. Then the node fairly adapts its own rate such that the capacity is not

exceed in the tree-based communication [7]. The spatial reuse effectively defines several wireless contention domains, in each of which the fair share may be different depending upon the number of flows traversing the domain. IFRC recognizes that the potential interferers of a node include nodes not just in the node's subtree or its neighbor's (parent included) subtrees, but also includes nodes in its parent's neighbor's subtrees. In other words, each node detects all flows from its potential interferers and allocates to each flow a fair and efficient share of its nominal bandwidth. For the allocation, IFRC adopts a max-min fair rate allocation.

4. Time Synchronization

Time synchronization is required for consistent distributed sensing and control. Common services in the SHM, such as coordination, communication, security, or power management, depend on the existence of global time. Besides, due to the delay of radio transmission or inherent internal sensor clock errors, the collected data in different sensors in the system may be unsynchronized. The time synchronization (TS) error can cause inaccuracy in health monitoring applications.

The timing-sync protocol for sensor networks (TPSN) periodically adjusts the offset of the clocks of nodes in two phases [8,9]. In the level discovery phase, a hierarchical topology in the network is created. The root node is assigned level 0 and broadcasts a level-discovery packet. Upon receiving the packet, the neighbors of the root node assign themselves level 1. Then, each level 1 node broadcasts its own level-discovery packet. This process is carried out throughout the network. After all, every node is assigned the level through the network-wide broadcasting. In the synchronization phase, the offset of the clocks is adjusted through the two-way message exchange between a pair of nodes. For example, node *A* sends a synchronization-pulse packet to node *B* at T_1 . The packet contains the level number of node *A* and the value of T_1 . Node *B* receives the packet at T_2 , where T_2 is equal to $T_1 + \Delta + d$. Here Δ and d represent the clock drift between the two nodes and propagation delay, respectively. At time T_3 , node *B* sends back an acknowledgement packet to node *A*. The acknowledgement packet contains the level number of node *B* and the values of T_1 , T_2 and T_3 . Node *A* receives the packet at T_4 and calculates the clock drift Δ and propagation delay d . With the drift, node *A* corrects and synchronizes its clock to node *B*. This process is carried out throughout the network and eventually every node is synchronized to the root node. In TPSN, due to a hierarchical topology formation and the two-way message exchange, it introduces a significant communication overhead and wastes energy and radio channel resources. Besides, TPSN suffers from the uncertainties of the overlapping transmission and reception time.

The flooding time synchronization protocol (FTSP) performs through one-way message exchange. It achieves TS by utilizing the MAC layer time-stamping and skew compensation with linear regression [10, 11]. For the offset adjustment by using MAC layer time-stamping, FTSP utilizes a $radz$ broadcasted message. The broadcasted message contains the sender's time stamp which is the estimated global time at the transmission. The receivers obtain the corresponding local time from their respective local clocks at message reception. The difference between the global and local time of a synchronization point estimates the clock offset of the receiver. If the local clocks had the exact same frequency, a single global-local time pair would be sufficient to synchronize two nodes. However, the frequency differences introduce clock drifts and this requires continuous resynchronization to keep the error in the micro-second range. Therefore, for the estimation of the receiver clock's drift and skew compensation, the linear regression prediction algorithm has been used. The linear regression prediction error is the difference between the global time given by one point and the estimated global time given by another point. Using the estimated error, the resynchronization interval is determined to acquire the desired precision.

5. Power Management

Wireless sensors typically are battery-powered, and their capacities are still limited. Once the battery has consumed all of its power, replacement of the battery located remotely can become a very expensive. Thus, power management is one of the biggest considerations in the SHM. There are three ways to manage power and prolong the lifetime: selecting low-power hardware, using power efficiently, and acquiring the power from a surrounding place. We focus on the power efficiency which is defined as appropriately utilizing limited battery power to maximize lifetime of WSNs.

[12] Enables two low-power monitoring modes - *Wait for Command* mode and *Event-triggering* mode. In other words, the microcontroller on a remote node features a low power sleep mode that can be exited via a watch-dog timer, or an external interrupt. In the *Wait for Command* mode, the microcontroller periodically awakes via a watchdog timer interrupt, turns on the telemetry hardware, and listens for a wake command from the base station. If it does not detect a wake command within the specified time duration, it returns to the sleep mode. In the *Event-triggering* mode, the microcontroller remains in the low power sleep mode until a rising external analog voltage triggers a hardware interrupt.

In [13], the sleep cycling and threshold triggering are designed for efficient power management in tree-based communication. In a deep sleep mode, the minimal power is consumed by making only the processor clock awake. To take advantage of the power savings of the deep sleep mode, while still allowing the gateway node access to the leaf nodes, the sleep cycling has been implemented. When the sleep cycling operates on the leaf nodes, they sleep for a set period of time and then wake up for a relatively short period of time, during which they can listen and receive message. The threshold triggering allows a subset of the leaf nodes to act as sentry nodes, in addition to their duties as leaf nodes. The sentry node collects data for a given period of time and determines if the threshold value has been exceeded. If the threshold is exceeded, the sentry node sends an alert message to the gateway node. Upon receipt of the alert, the gateway node makes the decision on the next actions to implement in the network.

For optimal power saving, the duty cycle of sleep mode and the threshold of the triggering event should be determined carefully and autonomously. Most commercial power management implementations focus on deciding when a node goes to sleep and how long the duty cycle would be through three policies – timeout policy, predictive policy, and stochastic policy [14].

- Timeout policy

It assumes that if the incoming request has an idle period that is longer than a threshold, then the idle period will be long enough to go to the sleep state. Timeout policy wastes energy during the threshold, as it keeps a node in the active state until the threshold passes. The selection of the threshold has to be done with a good understanding of both node's hardware characteristics and the typical request patterns. Although many implementations have a single fixed threshold, a number of studies have designed methods for adapting the threshold to changes in the requests with machine learning techniques or the distribution of idle periods [15, 16].

- Predictive policy

It attempts to predict the length of the next idle period by studying the distribution of request arrivals. Besides it attempts to transition a node into the sleep state with no idleness. When the prediction of timing and the length of the idle

period is correct, then the predictive policy provides a solution with no overhead. However, if the prediction is wrong, the potential cost can be quite large. The accuracy of idle period prediction is the keystone of the related studies [17,18].

- Stochastic policy

It is categorized into time-driven and event-driven which are based on the assumption that all distributions modeling the system are memory-less or that some distributions are history dependent. In addition, the policy can be stationary or non-stationary. A number of studies have proposed based on Markov chains [19, 20]. The system model using stationary discrete-time Markov decision process re-evaluates its decision periodically, so it adapts to the request arrival. However, since the decision is repeated periodically even when the system is idle, the power would be wasted. The system model using continuous-time Markov decision process saves more energy since there is no need to continually re-evaluate the policy in the low power state. However, it assumes that the exponential distribution to describe the system behavior, even though the exponential distribution is not fit in many practical cases.

Both timeout and predictive policies are a kind of heuristic policy. None of these policies can guarantee optimality. In contrast, policies that use stochastic models to formulate and solve the power management problem can provide optimal solutions within restrictive assumptions made when developing the system model.

6. Conclusion

In this paper, we surveyed the technical issues when applying a WSN for the SHM by focusing on the wireless communication aspects of a WSN. We introduced the software architecture for the SHM. Then, we identified three issues to successfully design a WSN for the SHM which are reliable data transfer, time synchronization, and power management. Most of the research works have focused on one of the three issues with an ideal assumption on the other issues or sometimes ignoring the effects of the other issues. However, these issues are tightly interconnected and the design choice of one issue affects the design of the others. Therefore, as a future works, an integrated optimal design process considering all the three issues needs to be devised to deploy more robust and efficient WSN for the SHM.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014043915, 2015010324).

References

- [1] G.-D. Zhou and T.-H. Yi, Recent developments on wireless sensor networks technology for bridge health monitoring. *Hindawi Mathematical Problems in Engineering*, vol. 2013, pp. 1-33 (2013).
- [2] S. Wijetunge, U. Gunawardana, and R. Liyanapathirana. Wireless sensor networks for structural health monitoring: considerations for communication protocol design. *Proceedings of International Conference of the IEEE Telecommunications (ICT'10)*, (2010) April 4-7; Doha, Qatar
- [3] P. Nie and Z. Jin. Requirements, challenges and opportunities of wireless sensor networks in structural health monitoring. *Proceedings of International Conference of the IEEE Broadband Network and Multimedia Technology (IC-BNMT'10)*, (2010) October 26-28; Beijing, China
- [4] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. *Proceedings of International Symposium of the ACM Information Processing in Sensor Networks (IPSN'07)*, (2007) April 25-27; Cambridge, US.
- [5] P. Wang, Y. Yan, G. Y. Tian, O. Bouzid, and Z. Ding, Investigation of wireless sensor networks for structural health monitoring, *Hindawi Journal of Sensors*, vol. 2012, pp. 1-7 (2012)

- [6] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: a reliable bulk transport protocol for multihop wireless networks. Proceedings of International Conference of the ACM Embedded Networked Sensor Systems (SenSys'07), **(2007)** November 6-9; Sydney, Australia
- [7] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. Proceedings of International Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'06), **(2006)** September 11-15; Pisa, Italy
- [8] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. Proceedings of International Conference on Embedded Networked Sensor Systems (SenSys'03), **(2003)** November 5-7; Los Angeles, US.
- [9] M. Bocca, L. M. Eriksson, A. Mahmood, R. J. antti, and J. Kullaa, A synchronized wireless sensor network for experimental modal analysis in structural health monitoring. Computer-Aided Civil and Infrastructure Engineering, vol.26, no.7, pp. 483-499 **(2011)**.
- [10] M. Maroti, B. Kusy, G. Simon, and A. L. edeczi. The flooding time synchronization protocol. Proceedings of International Conference on Embedded Networked Sensor Systems (SenSys'04), **(2004)** November 3-5; Baltimore, US.
- [11] T. Nagayama and B. F. Spencer Jr., Structural health monitoring using smart sensors, Newmark Structural Engineering Laboratory (NSEL) Report Series NESL-001, University of Illinois at Urbana-Champaign, Urbana, Ill, **(2007)**.
- [12] J. H. Galbreath, C. P. Townsend, S. W. Mundell, M. J. Hamel, B. Esser, D. Huston, ad S. W. Arms, "Civil structure strain monitoring with power-efficient high-speed wireless sensor networks," in Proceedings of International Workshop for Structural Health Monitoring, pp. 1-12, **(2003)**
- [13] J. A. Rice, K. Mechitov, S.-H. Sim, T. Nagayama, S. Jang, R. Kim, B. F. Spencer, Jr., G. Agha, and Y. Fujino. Flexible smart sensor framework for autonomous structural health monitoring. Smart Structures and Systems, vol.6, no.5-6, pp.423-438 **(2010)**
- [14] G. Park, T. Rosing, M. D. Todd, C. R. Farrar, and W. Hodgkiss, Energy harvesting for structural health monitoring sensor networks. Journal of Infrastructure Systems, vol.14. no.1, pp.64-79 **(2008)**
- [15] S. Irani, S. Shukla, and R. Gupta, Online strategies for dynamic power management in systems with multiple power-saving states. ACM Trans. Embedded Computing Systems, vol.2. no.3, pp. 325-346. **(2003)**
- [16] D. Ramanathan, S. Irani, and R. Gupta. Latency effects of system level power management algorithms. Proceedings of International Conference of IEEE/ACM Computer-Aided Design, **(2000)** November 5-9; San Jose, US.
- [17] L. Zhong and N.K. Jha. Dynamic power optimization for interactive systems. Proceedings of International Conference of VLSI Design, **(2004)** January 5-9; Mumbai, India.
- [18] T. Simunic and S. Boyd, Managing power consumption in networks on chips. Proceedings of Design, Automation and Test in Europe Conference and Exhibition, **(2002)** March 8-8; Paris, France.
- [19] L. Benini, G. Paleologo, A. Bogliolo, and G. DeMicheli, Policy optimization for dynamic power management. IEEE Trans. Computer-Aided Design, vol.18, no.6, pp.813-833 **(1999)**
- [20] Q. Qiu and M. Pedram. Dynamic power management of complex systems using generalized stochastic petri nets. Proceedings of Design Automation Conference, **(2000)** June 5-9; California, US.

