Efficient campaign-type structural health monitoring using wireless smart sensors

Jian Li,[a] Tomonori Nagayama,[c] Kirill A. Mechitov,[b] Billie F. Spencer, Jr.[a]

[a]Department of Civil and Environmental Engineering,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

[b]Department of Computer Science,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

[c]Department of Civil Engineering, University of Tokyo, Tokyo 113-8656, Japan

ABSTRACT

Wireless Smart Sensor Networks (WSSNs) have attracted great attention in recent years for Structural Health Monitoring (SHM), enabling better understanding of the dynamic behavior of large scale civil infrastructures through dense deployment of sensors. With a fraction of the deployment time and cost compared with wired SHM systems, WSSNs can serve as ideal systems for campaign-type monitoring for (i) short-term, in-service performance evaluation, (ii) post-disaster condition assessment, (iii) design optimization of long-term SHM system before permanent deployment, etc. Efficient data collection is generally needed in campaign monitoring due to limited operation time. A number of improvements have been made to the Illinois SHM Project (ISHMP) Services Toolsuite to facilitate efficient data collection for campaign monitoring. A post-sensing time synchronization scheme is proposed to reduce the latency of data collection while maintaining high accuracy of synchronization of collected data. A multi-hop bulk data transfer approach using multiple RF channels is also implemented to achieve high data throughput.

Keywords: wireless smart sensors, campaign-type health monitoring, time synchronization, multi-hop communication

1 INTRODUCTION

Wireless Smart Sensor Networks (WSSN) have attracted great attention in recent years in Structural Health Monitoring (SHM), by enabling better understanding of the dynamic behavior of large scale civil infrastructures through dense instrumentation.[1] Traditional wired SHM systems are generally expensive and time-consuming to install due to cabling, and are likely to suffer from data inundation. With the wireless communication and on-board processing capabilities, wireless SHM systems have the advantages of being low cost, easy to deploy and maintain and the potential of reducing data inundation through careful design of software.[2] Therefore, WSSNs can serve as ideal systems for campaign-type SHM such as: (i) short-term in-service performance evaluation, (ii) post-disaster condition assessment, and (iii) design optimization of long-term SHM system before permanent deployment, etc.

Minimizing the time for data collection is critical for campaign-type SHM. For example, when a monitoring effort requires stopping traffic, efficient data collection allows the impact to be reduced or more data sets to be measured given a fixed operation time. Also, after major disasters such as earthquakes, tsunami, or typhoon, evaluating the capacity of civil infrastructures such as bridges, buildings, and railways must be done in a timely manner to minimize the impact on society. In addition, sensing should start quickly for transient events, to avoid missing the event. Monitoring of the Government Bridge, a swing bridge in Rock Island, Illinois, provides an example of such a transient event. The swing event usually occurs over only a few minutes; preparation for sensing must be kept to a minimum to ensure capturing the event.

The time for data collection in wireless SHM systems is mainly determined by three steps: time synchronization, sensing, and data transport. Among them, sensing time is a constant if the sampling frequency and number of data points

to be collected are determined by application specific requirements. However, time consumed by the other two steps, i.e. time synchronization and data transport, can be reduced. Time synchronization is the process by which the local clocks are synchronized on each sensor node in the network. Accurately synchronized clocks are essential for data collection in wireless SHM; errors in phase information impacts estimation of the structural mode shapes, which are critical for damage detection.[4] To compensate for clock drift, Nagayama and Spencer [4] proposed a strategy that estimates the drift rate before sensing using a least-squares linear regression approach. However, this strategy introduces latency in the start of sensing, which increases the total data collection time and may also cause some important events to be missed. Moreover, for long term data collection, the clock drift rate may change due to temperature changes. Relying on the clock drift rate estimated before sensing may introduce large synchronization errors.

Data transport is time consuming for the large amounts of data that are typical in SHM applications, especially when multihop communication is required.[6] Appropriate coordination and scheduling are the keys to avoiding congestion and thus reduce data transport time. In the Illinois SHM Project (ISHMP) Services Toolsuite,[5] a general purpose AODV routing protocol is implemented and the data transport in the network is done in a sequential manner, i.e. the gateway node sends requests to leaf nodes one by one for data retrieval. This strategy utilizes only one radio channel so that only one pair of sensor nodes are transporting data at any time. Nagayama et al. [6] proposed a multihop bulk data transfer approach which utilizes multiple Radio Frequency (RF) channels such that multiple pairs of sensor nodes can deliver data simultaneously. The network throughput is significantly improved in the latter approach.

In this paper, a number of improvements to the ISHMP Services Toolsuite to facilitate efficient data collection for campaign monitoring are presented. A post-sensing time synchronization scheme is proposed to reduce the latency of data collection and provide more effective clock drift compensation for long term data collection. Test results show that high accuracy synchronization is maintained for both short and long term data collection. In addition, removing outliers and using nonlinear curve fitting are found to be essential for effective clock drift compensation. The multihop bulk data transfer approach using multiple RF channels [6] is also integrated into the ISHMP Service Toolsuite. The implementation and integration details and a preliminary test result are also presented. The paper concludes with a discussion of future development plans.

2 A POST-SENSING TIME SYNCHRONIZATION SCHEME

In this section, the current synchronized sensing middleware service implemented in the ISHMP Service Toolsuite is introduced, followed by a new post-sensing time synchronization scheme which aims to reduce the latency for starting sensing by moving the clock drift compensation phase after sensing is finished. This approach also allows for more effective clock drift compensation for long-time data collection, during which the clock drift rate may change due to temperature variations. Implementation details such as outlier detection and nonlinear curve fitting for drift compensation are discussed herein. Finally, the experimental test results of the post-sensing time synchronization scheme are presented.

2.1 Current synchronized sensing middleware service in the ISHMP Service Toolsuite

The synchronized sensing middleware service in the ISHMP Service Toolsuite has two components, including clock synchronization and data synchronization. The former is realized by a hierarchical synchronization tree protocol based on periodic beacon messages using the Flooding Time Synchronization Protocol (FTSP) [8] packet timestamping method. Clock drift is considered to be linear, and the drift rate is estimated by a least-squares approach with the beacon data points. The estimation is performed before sensing and takes 30 seconds to collect a sufficient number of data points. However, synchronized clocks do not guarantee synchronized data due to the uncertainty of start-up time, the difference of sampling frequency among sensor nodes and the fluctuation of sampling frequency over time.[4] Data synchronization is achieved by a resampling based approach, which addresses the above three issues towards synchronized data.[4] The procedure is illustrated in Fig. 1(a).

2.2 Post-sensing time synchronization

As shown in Fig. 1(a), the current approach to clock drift rate estimation imposes a 30-second delay before sensing starts. To reduce the latency and improve the efficiency of data collection, a single beacon/synchronization message is used to calculate the initial clock offset (Δt_0) and correct the local clocks of the leaf nodes; subsequently, all sensor nodes

start sensing at roughly the same time. Meanwhile, the gateway node keeps broadcasting beacon messages with its global time stamps (t_{gb}) during sensing. Upon receiving these beacon messages, the leaf nodes record the local time stamps (t_{lb}) and calculate the offset (Δt_b). After sensing is finished, the vector of clock offsets and local time stamps are used to estimate the clock drift rate over time using regression analysis, which is then used to correct the time stamps of sensor data (t_{gd}). Resampling is finally carried out based on the drift compensated time stamps. The procedure is illustrated in Fig. 1(b) and Fig. 5.

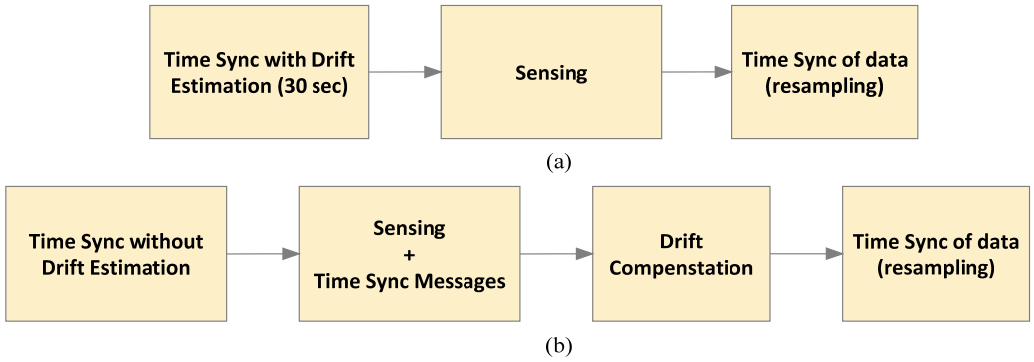
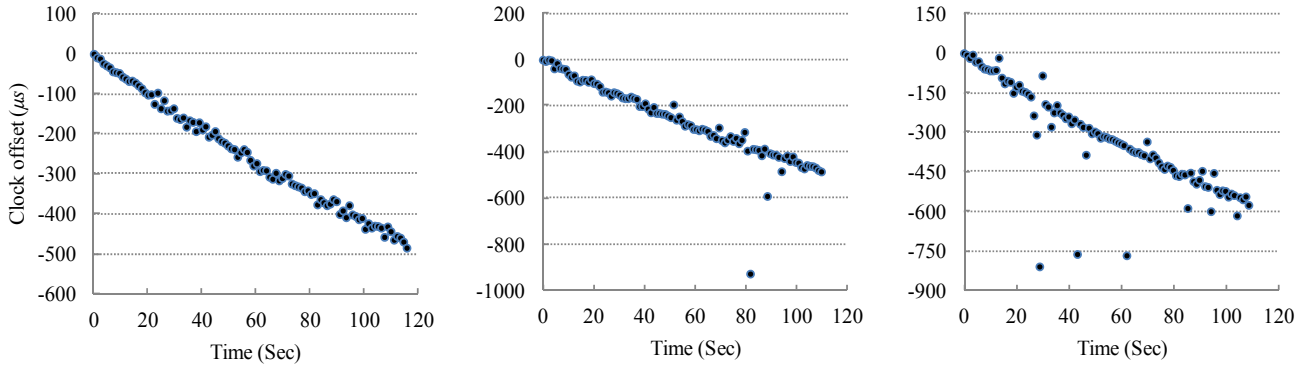


Figure 1. Synchronized sensing procedures: (a) pre-sensing time synchronization; (b) post-sensing time synchronization.

2.2.1 Outlier detection and removal using Cook's Distance

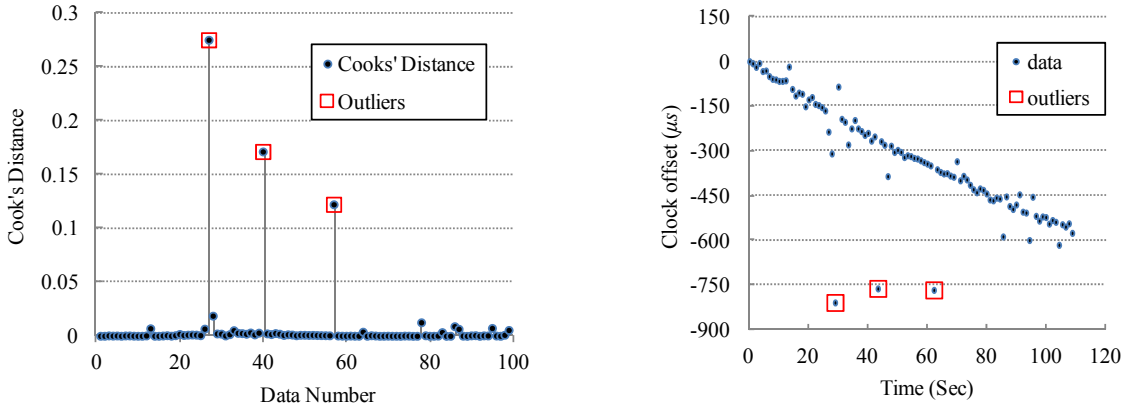
One issue to be addressed for post-sensing time synchronization is the potential conflict between the sample acquisition and RF communication for synchronization messages. The proposed post-sensing time synchronization approach requires the smart sensors to handle these two tasks concurrently. With the concurrency model of TinyOS, all tasks are scheduled and executed in a First In, First Out (FIFO) manner. If the entire computation related to sending/receiving messages is finished in the window between sample timestamping, there is no conflict; however, if the message processing is delayed so it overlaps with the sample acquisition and timestamping code, it's up to the scheduler to determine which one executes first and which one is delayed.

Tests were carried out to evaluate the effect of such a conflict. The Imote2 sensor platform with the SHM-A sensor board (commercially available as ISM400 from MEMSIC) [9-10] was used in the tests. 100 seconds of data were collected at three different sampling rates: 25 Hz, 100 Hz and 280 Hz. Meanwhile, for all three cases, 100 beacon messages were sent during sensing. The clock offsets collected over time are plotted in Fig. 2. Increasing the sampling frequency is shown to increase the chance for outliers. One can imagine that with even higher sampling rates, the number of outliers will increase and can affect the accuracy of clock drift rate estimation. Therefore, detecting the outliers and removing them before regression analysis is essential for effective drift compensation.



(a) f_s = 25 Hz (b) f_s = 100 Hz (c) f_s = 280 Hz

Figure 2. Clock offset during sensing under different sampling rates.



(a) Cook's distances (b) Outliers

Figure 3. Outlier detection using Cook's Distance.

Cook's distance [11] is employed in this study to detect outliers from the collected synchronization data. In regression analysis, Cook's distance measures the influence of removing one certain data point, which is calculated as

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{y}_j - \hat{y}_j (\mathrm{i}) \right)^2}{(p+1)s^2}, \quad i = 1, 2, \ldots, n \tag{1}$$

in which $\hat{y}_j$ and $\hat{y}_j(\mathrm{i})$ are the regression estimates of the observation j using the full data set and the data set with the i^{th} data point removed, respectively; s^2 is the mean square error of the regression model; $p+1$ is the number of parameters in the regression model. Large value of Cook's Distance indicates outlier in the synchronization message caused by the conflict. As a rule of thumb, a data point is treated as outlier when

$$D_i \geq \frac{4}{n-(p+1)} \tag{2}$$

Cook's Distance is implemented in TinyOS as a numerical service in the ISHMP Services Toolsuite. Fig. 3(a) shows the Cook's Distance for the data given in Fig. 2(c), and the detected outliers are highlighted in Fig. 3(b), which demonstrates Eq. (2) effectively removes the outliers from the collected time synchronization data.

2.2.2 Nonlinear regression analysis for clock drift compensation

Clock drift is another common issue that needs to be addressed for time synchronization of distributed systems. In the current ISHMP synchronization protocol, clock drift is compensated by estimating a constant drift rate using linear regression. However, the crystal oscillator in the timer circuit of Imote2 is sensitive to temperature changes.[12-13] Clock drift usually is linear if the temperature remains constant. However, the temperature of the Imote2 oscillator could change due to (1) environmental temperature change during sensing and (2) the heat generated by the Imote2 CPU and the Quickfilter ADC chip on the SHM-A sensor board. Fig. 4 shows the results from an indoor test with four Imote2 leaf nodes. In the test, an application was designed to collect time stamps and temperature readings from these leaf nodes during sensing. While the indoor environment provides constant ambient temperature, the on-board temperature of Imote2 was raised by almost 6 ℃ in 10 minutes due to the heat generated by the CPU and the Quickfilter chip on the SHM-A sensor board. As a result, all four curves of clock drift have significant nonlinearity (see Fig. 4). Note that unit-to-unit variation is found in the clock drift between sensors, which is why the sensors exhibit different offset curves for very similar temperature changes. Therefore, for long term data collection during which temperature change due to either of the above two factors may become significant, nonlinear regression analysis is needed to achieve high accuracy of time synchronization. The clock drift compensation procedure employing both the Cook's distance and nonlinear regression analysis is illustrated in Fig. 5.

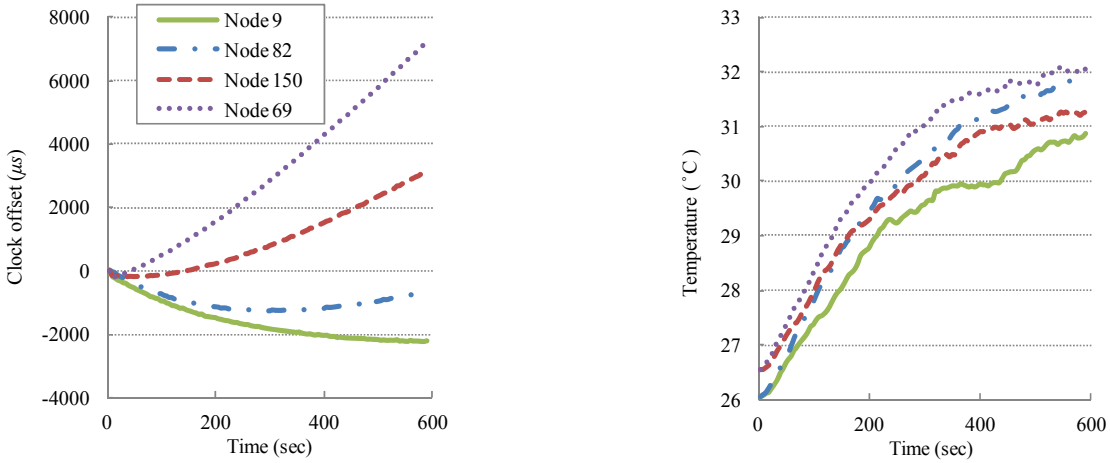


Figure 4. Nonlinear clock drift (left) due to temperature change (right).

1. Record beacon data (during sensing): $\Delta t_b = t_{gb} - t_{lb}$ (3)

2. Outlier detection and removal for beacon data (after sensing)

3. Regression analysis (after sensing): $\Delta t_b = c_0 + c_1 t_{lb} + c_2 t_{lb}^2 + \cdots + c_p t_{lb}^p$ (4)

   ($c_0, c_1, \ldots c_p$ are regression parameters; p is the order of regression analysis.)

4. Correct data time stamps (after sensing): $t_{ld} = t_{gd} - \Delta t_0$ (local time stamp of sensor data) (5)

   $\Delta t_{ld}^{corrected} = c_0 + c_1 t_{ld} + c_2 t_{ld}^2 + \cdots + c_p t_{ld}^p$ (6)

   $t_{gd}^{corrected} = t_{ld} + \Delta t_{ld}^{corrected}$ (7)

Figure 5. Clock drift compensation procedure in the post-sensing time synchronization scheme.

2.3 Time synchronization accuracy

To estimate the performance of the proposed post-sensing time synchronization scheme, as depicted in Fig. 6, tests were conducted using one Imote2 sensor platform as the gateway node connected to the base station computer and three Imote2s with SHM-A sensor boards as leaf nodes. A Siglab spectrum analyzer [14] provides the signal to the 4th channel of the SHM-A sensor board on each Imote2. To make sure each sensor node receives an identical signal from Siglab, split connectors were used to connect all senor nodes to the same output channel of Siglab. The input signal is Band Limited White Noise (BLWN) with 20 Hz bandwidth. The sampling frequency of data acquisition is 100 Hz. Time synchronization error between two sensor nodes can be evaluated through the phase angle of the cross power spectrum between their acceleration data. More specifically, linear curve fitting is applied to the phase angle in the range between 0 and 20 Hz to find the slope of the phase angle, θ, which can be converted to time synchronization error in microseconds as

$$TS_{error} = \frac{\theta}{2\pi} \times 10^6 \ (\mu s) \quad (3)$$

Several tests were carried out for three different sensing durations, including 1 minutes, 10 minutes and 30 minutes. Both linear curve fitting and nonlinear curve fitting (5th order) were applied to clock drift compensation. Outlier detection and removal algorithm was used for both cases. In the calculation of the cross power spectral densities, the Hanning window with the width of 1024 Fast Fourier Transform (FFT) points and 50% overlapping was adopted. For each pair of sensor nodes, the absolute values of the synchronization errors from three repeated tests were averaged. The results are summarized in Table 1. The proposed post-sensing time synchronization scheme achieves high accuracy both in short and long term data collection. Moreover, for long term data collection, especially for the 30-minute case, when the proposed nonlinear curve fitting is applied in the clock drift compensation, the accuracy of time synchronization is improved significantly compared with the case with linear curve fitting.

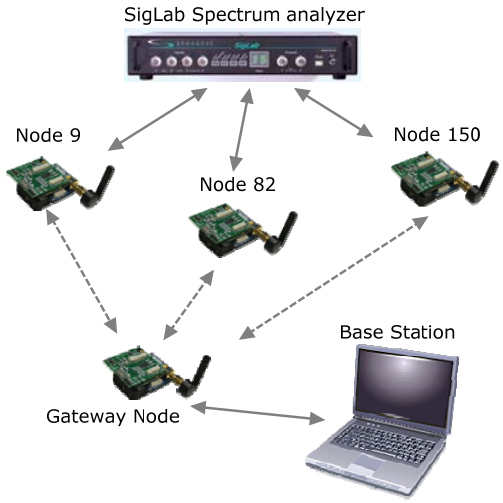


Figure 6. Test Setup for time synchronization accuracy estimation.

Table 1. Time Synchronization errors with post-sensing time synchronization.

Sensing Duration	Curve Fitting Approach in Drift Compensation	Pairwise synchronization Error (μ sec)		
		Node 9 and 82	Node 9 and 150	Node 82 and 150
1 min	1st order	26.3	6.8	22.8
	5th order	25.5	6.7	22.6
	difference*	-2.98%	-1.21%	-0.87%
10 min	1st order	19.2	10.6	25.2
	5th order	18.4	10.4	24.8
	difference*	-4.25%	-2.05%	-1.57%
30 min	1st order	11.3	15.0	26.3
	5th order	6.9	13.0	19.9
	difference*	-38.55%	-13.37%	-24.15%

* difference = (5th order - 1st order)/(1st order)*100%

3. MULTIHOP BULK DATA TRANSFER

In multihop communication, transporting large data set in wireless SHM applications is challenging. In the current ISHMP Services Toolsuite, to avoid packet collisions, data retrieval in multihop is done in a sequential manner, i.e. the gateway node requests data from each leaf node one by one. This approach slows down the data retrieval significantly especially when the number of hops is large. Nagayama et al. [6] proposed a multihop bulk data transfer protocol which utilizes multiple RF channels to allow multiple neighboring pairs of nodes to transmit data simultaneously. As a joint research effort between the University of Illinois and the University of Tokyo, multihop bulk data transfer has been integrated into the ISHMP Services Toolsuite. For completeness, the background of the protocol is first introduced,

followed by some additional considerations to improve the efficiency of data collection in campaign monitoring using the multihop bulk data transfer protocol.

3.1 Multihop Routing for single sink data collection

To facilitate the multihop bulk data transfer with multiple RF channels, a specific routing protocol was designed.[14] A single sink node is assumed, which in this case is the gateway node in SHM system. The routing process is performed in a backwards manner from the leaf nodes to the gateway node. First, the leaf nodes within one-hop range of the gateway node establish routes; then the leaf nodes that are two-hops away from the gateway node find the routes. The process repeats for a predetermined number of times until all nodes in the network have a route to the gateway node. The procedure is illustrated in Fig. 7.

To initialize the routing process, the gateway node sends a command to the leaf nodes. Upon receiving the command, the leaf nodes start to broadcast the Route Request (RREQ) messages with Time-To-Live (TTL) equal to one hop. To avoid heavy package collision when many nodes start to broadcast messages, each node waits for a random interval before broadcasting. The value of TTL decides how far the RREQ messages can travel. Having TTL = 1 reduces the packet collision in the network and allows the shortest path to be built. When the gateway node receives the RREQ messages, it examines the Received Signal Strength Indicator (RSSI) values and responds with Route Reply (RREP) messages if the RSSI value is higher than a predetermined threshold value. When the RREP messages reach the leaf nodes which sent the RREQ messages, the leaf nodes examines the RSSI value and establish routes to the gateway node if the RSSI value is higher than the threshold. After the first stage, all nodes within one-hop range from the gateway node have established routes to the gateway node.

After waiting for a random interval, the rest of the leaf nodes that still do not have routes to the gateway node continue to broadcast RREQ messages. Now not only the gateway node, but also the leaf nodes which have built routes to the gateway node are eligible to reply with RREP messages. If the RSSI values in both the RREQ and RREP messages satisfy the threshold, the routes are established. The same process is repeated for nodes that are more hops away until all nodes in the network have routes to the gateway node or a timeout timer is fired.

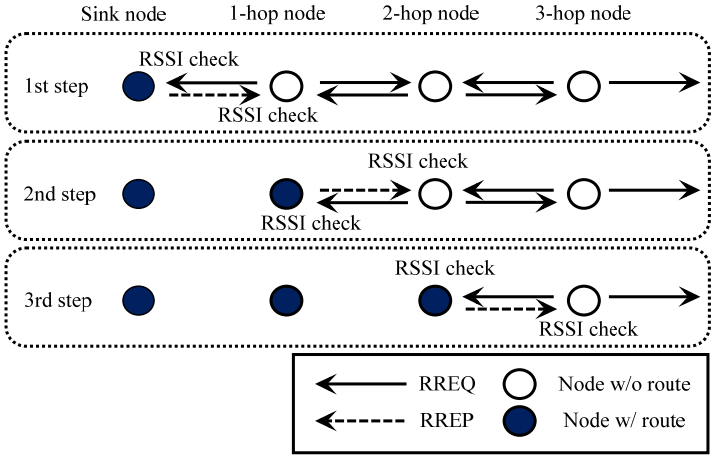


Figure 7. Multihop routing process for single sink data collection (Nagayama et al. [6]).

3.2 Multihop data forwarding using multiple RF channels

Having the routing table obtained in the previous step, the network can be divided into layers assigned with two additional channels besides the common communication channel. One is the data transmit channel and the other is the data reception channel. The data reception channel of the lower layer nodes is the same as the data transmit channel of its upper layer nodes. For example, the nodes which are *n*-hops away from the gateway node, also called the *n*-hop layer nodes, have the data reception channel the same as the data transmit channel as the (*n*+1)-hop layer nodes. The IEEE802.15.4 RF device employed by the Imote2 sensor platform provides 16 user-selectable and non-overlapping

channels. These 16 channels are used in a round robin manner in the case when more than 16 hops are needed, providing network scalability. For example, the 17-hop layer nodes and the 1-hop layer nodes will share the same channels; however, the interference between these two layers is considered minimal because they are physically far apart from each other.

When sensing is finished and data is saved into the flash memory of each leaf node, the gateway node sends a command to the leaf nodes to initiate the data forwarding process. When data is ready to be transmitted back to the gateway node, all nodes in the network switch to their own data reception channel from the common communication channel. After a random wait interval, when a sender node starts to transmit data using the routing table, it switches its RF channel to its data transmit channel, and examines whether the channel is being used by other nodes. If the channel is being used, the sender node waits for another random interval. If not, the sender node sends an inquiry packet to the receiver node asking if the receiver node is ready. If the receiver node is not communicating with other nodes and has adequate buffer space, the data forwarding begins. Using a reliable data transfer service,[4] the sender sends all sensor data stored in flash memory to the receiver and then switches its RF channel to the data reception channel. If the receiver is not ready, the sender waits for a random interval before starting the process again. Fig. 8 illustrates the simultaneous data transfer between multiple pairs of sensor nodes within the network using multiple RF channels.

3.3 Preliminary test results

In a preliminary test results reported by Nagayama, et al.,[6] 49 sensor nodes were installed on a suspension bridge in Japan. Routes with maximum of eight hops were established in the network. Data was collected using two gateway nodes. One of the gateway nodes collected data from 24 sensor nodes; each one has 108 kB of data. Six minutes was used to retrieve all data back, resulting in 7.2 kB/s throughput. Compared with the data throughput of the MintRoute-based approach reported by Kim, et al.,[16] which is about 0.75 kB/s, the proposed Multihop Bulk Data Transfer approach achieved approximately a tenfold improvement in performance.

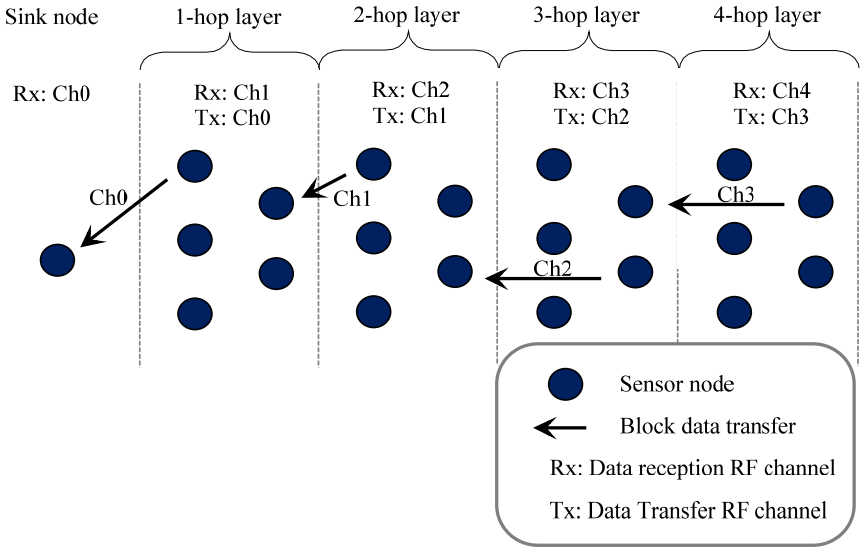


Figure 8. Bulk data transfer using multiple RF channels (Nagayama et al. [6]).

3.4 Implementation in the ISHMP Services Toolsuite

The Multihop Routing and Data Forwarding methods described above are modularized into two separate interfaces in the ISHMP Service Toolsuite, which are then integrated into the current *RemoteSensing* application for data retrieval once data becomes available. Meanwhile, current command delivery approach does not suit the need to maintain high efficiency of the new Multihop Bulk Data Transport method; a different flooding based command delivery approach is therefore implemented.

In the current ISHMP Services Toolsuite, commands are delivered using a series of unicasts based on the reliable communication protocol. This acknowledgement based method ensures the reliability of command delivery but may slow down the application when the network size is large and a few nodes do not respond, thus cancelling out the benefit of the Multihop Bulk Data Transport method. Moreover, the Data Forwarding service assumes all sensor nodes start forwarding data approximately at the same time. Therefore, commands, such as the initialization of multihop routing and data forwarding, are all delivered using flooding.

Flooding is also implemented to wake up the network. For campaign monitoring, to conserve energy during sensor node installation, sensor nodes are put into sleep cycle, in which the nodes spend most of the time in a low-power deep sleep mode and wake up periodically to listen for external commands. Waking up the network as quickly as possible is also critical for efficient data collection. In the implementation, to wake up the network, the gateway node keeps broadcasting the wakeup messages to the network for duration t, which is the sleep time in the sleep cycle. Upon reception of the wakeup message after switching to the listening mode, the leaf nodes stay awake and also keep broadcasting the same wakeup command for duration t. Meanwhile, the gateway node keeps listening to the wakeup messages coming back from the network during the flooding process. When the gateway node no longer receives wakeup messages from the network, the wakeup process is considered completed. If the maximum number of hops in the network is N, after $N \times t$ seconds, it is almost sure that all nodes are awake.

One more issue is resetting the network after data forwarding is finished. As described in section 3.2, all nodes will switch back to their own data reception RF channel after current data block has been transferred, waiting for the next round of data transfer. Without a notification message sent from the gateway node or a timer to timeout the data forwarding, the leaf nodes will not know whether the data forwarding is completed or not and stay in their own RF reception channels, making the network unreachable by communication within any one RF channel. In the implementation, the gateway node sends the reset command by scanning all the available RF channels. Upon the reception of the reset command, leaf nodes also broadcast the command by scanning all the RF channels. After a timeout timer gets fired, all nodes that have received the reset command reset themselves and hence switch back to the common communication RF channel.

4. FUTURE ENHANCEMENTS

Successful integration of the two components described in the preceding sections, i.e. the post-sensing time synchronization and multihop bulk data transport services, have enabled efficient campaign-type wireless SHM system. This section discusses ongoing developments to further enhance these services.

1. The post-sensing time synchronization service needs to be extended to the multihop case. In current implementation, to reduce the interference with data acquisition, the synchronization messages are allowed to travel only one hop, hence limiting its applicability to single-hop networks. To extend the service to multihop case, one solution is to utilize the routing table established in the multihop routing stage to forward the synchronization messages to next hops.

2. To support the above extension, the routes in the forward direction, i.e. from the gateway to leaf nodes, need to be built. Since only the RREQ senders establish routes, the routing information is not available in the forward direction. To build routing table in the forward direction, after establishing a route at a RERQ sender, the sender can send a packet to the RREP sender informing the established route.

3. The RSSI-based link quality estimator may not be adequate in some environments. The routes with low RSSI value will not be built under current implementation. Other link quality measures, which also take into account the Link Quality Indicator (LQI) can be used for more reliable link quality estimation.

4. The integrated system needs to be tested with a real world application. The authors have a plan to deploy a dense array of sensors to the Government Bridge[3] to test the performance of the proposed campaign-type wireless SHM system.

5. CONCLUSIONS

Critical components of an efficient campaign-type SHM system based on WSSNs have been proposed and implemented. A post-sensing time synchronization scheme was proposed to reduce the latency of data collection while maintaining high accuracy of synchronization of collected data. Results showed that high accuracy of synchronization was maintained for both short and long term data collection. In addition, removing outliers and nonlinear curve fitting were found essential for effective clock drift compensation. A multi-hop bulk data transfer approach using multiple RF channels was also implemented to achieve high data throughput. Future enhancements were discussed in order to achieve an integrated and highly efficient wireless SHM system for campaign monitoring.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the partial support of this research by the National Science Foundation Grants CMS 09-28886 and NSF CPS 10-35773, and the China Scholarship Council which partially supports the first author's PhD study.

REFERENCES

[1] Spencer Jr., B. F., Ruiz-Sandoval, M. and Kurata, N., "Smart Sensing Technology: Opportunities and Challenges," *Structural Control and Health Monitoring* 11, 349–368 (2004).

[2] Rice, J. A., Mechitov, K., Sim, S. H., Nagayama, T., Jang, S., Kim, R., Spencer Jr., B. F., Agha, G. and Fujino, Y., "Flexible Smart Sensor Framework for Autonomous Structural Health Monitoring," *Smart Stuct. Syst.* 6, 423-38 (2010).

[3] Giles, R. K., Kim, R., Spencer Jr., B. F., Bergman, L. A., Shield, C. K. and Sweeney, S. C., "Structural Health Indices for Steel Truss Bridges," T. Proulx (ed.), *Civil Engineering Topics, Volumn 4*, Conference Proceedings of the Society for Experimental Mechanics Series, 7, 391-398 (2011).

[4] Nagayama, T., and Spencer Jr., B. F., "Structural Health Monitoring using Smart Sensors, " NSEL Report Series, No. 1, University of Illinois at Urbana-Champaign (2007). http://hdl.handle.net/2142/3521.

[5] Illinois Structural Health Monitoring Project (ISHMP) Service Toolsuite. http://shm.cs.uiuc.edu/

[6] Nagayama, T., Moinzadeh, P., Mechitov, K., Ushita, M., Makihata, N., Ieiri, M., Agha, G., Spencer Jr., B. F., Fujino, Y., Seo. J., "Reliable multi-hop communication for structural health monitoring," *Smart Structures and Systems*, vol. 6, no. 5, pp. 481-504 (2010).

[7] Nagayama, T., Sim, S.H., Miyamori, Y. and Spencer Jr., B. F., "Issues in Structural Health Monitoring employing smart sensors," *Smart Structures and Systems*, 3(3), 299-320 (2007).

[8] Maroti, M., Kusy, B., Simon, G., and Ledeczi, A., "The flooding time synchronization protocol." *Proceedings of 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, 39-49 (2004).

[9] Rice, J. A. and Spencer Jr., B. F. "Flexible Smart Sensor Framework for Autonomous Full-scale Structural Health Monitoring," *NSEL Report Series,* No. 18, University of Illinois at Urbana-Champaign (2009). http://hdl.handle.net/2142/13635.

[10] MEMSIC, Inc. "ISM400, Imote2 Structural Health Monitoring Board", Andover, MA (2010).

[11] Cook, R. Dennis, "Detection of Influential Observations in Linear Regression". *Technometrics,* 19 (1): 15–18 (1977).

[12] Uddin, M.B. and Castelluccia, C., "Toward clock skew based wireless sensor node services," In *ICST Wireless Internet Conference (WICON)'10*, 1–9 (2010).

[13] Yang, Z., Cai, L., Liu, Y., and Pan, J., "Environment-Aware Clock Skew Estimation and Synchronization for Wireless Sensor Networks," *IEEE Infocom*, Orlando, FL, USA, March (2012).

[14] Spectral Dynamics, Inc. <http://www.spectraldynamics.com> (Feb. 2, 2012).

[15] Perkins, C. E. and Royer, E. M., "Ad-hoc on-demand distance vector routing", *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, February (1999).

[16] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S. and Turon, M., "Health monitoring of civil infrastructures using wireless sensor networks", *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, Cambridge, Massachusetts, USA, April (2007).