A Reliable Transfer Protocol for Multi-Parameter Data Collecting in Wireless Sensor Networks

Xiaotian Fei*, Qingbo Zheng*, TaoTang*, Yuzhi Wang*, Pengjun Wang*, Wei Liu*, Huazhong Yang*

*Department of Electronic Engineering, Tsinghua University, China

fxt10@mails.tsinghua.edu.cn, zhengqb10@mails.tsinghua.edu.cn, tt07@mails.tsinghua.edu.cn

Abstract—**A reliable data collection protocol for structural health monitoring (SHM) is presented. This application is deployed on Ronghu Bridge in the city of Wuxi, Jiangsu Province, China. The system monitors seven types of sensor (vibration, strain, tension, temperature, bridge deck settlement, bridge tower tilt, wind speed and direction) in real time. Since radio transmission consume most part of the node battery power, this system improves the battery life by optimizing the radio duty cycle. Time synchronization is adopted for time slice control, where each slice consists two parts, working and sleeping. Every working period is able to handle different sensor data and each time slice is adjustable according to different data type. For reliable data collection, the gateway node send nack request at the end of the time slice for retransmission. This application has been developed on TinyOS environment, and tested first on our test bed in the lab then on Ronghu Bridge with estimated operation time over one year without battery replacement. The result of this protocol shows better performance over CTP (TinyOS supported) protocol.**

Keywords— **Wireless Sensor Networks, Structural Health Monitoring, transfer, Reliable collection**

I. INTRODUCTION

It is very important to maintain the health of a bridge structure while it is in operation. Currently, only a few bridges installed real-time structural health monitoring (SHM) system. In most cases, people inspect visible damages weekly and check the whole structure manually with expensive equipment every few years. A SHM system not only can be used to check the long-term structural changes, but also able to detect structural damage in real-time. Wireless sensor networks (WSNs) technology is very suitable in the structural health monitoring applications. Compare to conventional SHM system, using WSNs for SHM can lower the cost of deployment and maintenance, and the network can be easily expanded for future upgrades.

Large structure such as bridge is a very complex system, one or two parameters is not enough for analysis. Study [1]-[4] show the possible solution to monitor several types of sensor for a structural. The Golden Gate Bridge monitors only the vibration data [1]. The Jindo Bridge monitors several types of data with a simple wireless network [4].

Based on application shown in [5], we improved the system with seven types of sensor, together they can describe the whole picture of the bridge we are monitoring. Different sensors use different transmission pattern, we divide those sensors into two categories for study. The first category sensors send one packet at interval of a few minutes, such as steel structure strain, temperature, bridge deck settlement, bridge tower tilt, wind speed and direction. The second category sensor samples vibration data with a frequency of 50-200Hz. This sampling frequency is faster than the multi-hop network throughput. Therefore, data are saved to local non-volatile memory and can be sent until network is not busy.

WSNs technology has been developed in the academy world, many challenges are still not solved, especially in the real world applications. One key problem is the node power consumption; WSN nodes are often deployed with battery power, they have to work in a low power state for longer network operation life. In general, the RF power consumes most energy of a node. The Micaz [6] node (one TinyOS [7] supported WSN nodes) uses CC2420 [8] RF chip, the maximum radio current consumption for transmit is 17.4mAh. Lower the RF transmitting power also means shorter the communication distance between two nodes. Therefore, the network needs to be multi-hopped for data transmission. On the other hand, CC2420 consumes 18.8mAh power in receiving state. Thus, reduce the RF working time could save valuable energy. In the following discussion, we use "work time" when RF chip is turned on, and "sleep time" when it is off.

As mentioned above, the system needs to transmit varies types of sensor data, and save power as much as possible at the same time. We propose a WSN system to meet the above requirements. The system divides time into time slices for data transmission. Time slices are synchronized for network operation. Each time slice contains work time and sleep time, the beginning of each time slice is the work time where nodes wake up and exchange data, they go to sleep after the work is complete. Each time slice can perform different tasks, for example, a time slice can be used to send the first category data and another time slice for vibration data transmission. Before the end of each time slice, the sink node uses NACK strategy to ensure that all packets have been collected.

The advantages of time slice are show in the follows:

- Regular sleep can reduce the duty cycle, and increase the system life time.
- System is able to handle multiple sensor data type in the transmission.

The following paper is organized as follows. Related work of WSNs system designs are shown in section II. The proposed system design is described in section III. Lab evaluation and field deployment are described in section IV and V. And conclusion is shown in section VI.

II. Related Works

In the actual deployment, the wireless signal environment is very different. For different environments, each project has its own method [9]. Application deployed at the Golden Gate Bridge [1], [10] uses Straw system that samples vibration data at multiple locations. This system uses directional antennal for a 46-hop communication. However this system does not support multiple nodes to transmit at the same time.

Flush [11] is an improved system; it assumes the same communication environment as above, which is very good. The algorithm is improved to achieve a better transmission rate. However, the link-transmission success rate assumed by this system is 99.9%, which is not possible for all the applications.

RCRT [12] allows nodes to send data at their will, but use sink node to assign the transmission rate of each node. This protocol is deployed on the wild for birds' nests monitoring. This system cannot achieve a very high data throughput.

Koala [13] reduces the duty cycle to 0.1% to save the power. The network code is less complex and it uses flexible routing calculation to control the data path.

CTP [14] is a simple collecting protocol that no confirmation is made at sink, but the protocol is very lightweight and can be install for less complex systems.

PIP has a "Multiple radio channels, TDMA-based MAC" [15], and designed for high throughput bulk transfer. It focuses on MAC layer, but uses relatively complex WSN nodes.

JRPA [16] uses a distributed algorithm that makes effort on network lifetime maximization while considering the routing layer, power control in the physical layer and link access in the MAC layer. The simulation result is very well, but it is lack of actual network deployment.

Overall, each protocol has its own particular scene, for different communications environment, we have to take a different approach.

III. Design

Figure 1. Ronghu Bridge over view

The designed system is for an upgrade at Ronghu Bridge, Wuxi, Jiangsu, China. The original implementation is shown in [5]. The Ronghu Bridge is a steel box girder cable-stayed bridge.

The designed system is able to adapt for future bridge applications with minimum software change.

A. Wireless Signal Test

Before design the embedded software system, we test the communication quality carefully on the bridge. There are two places we had to test: on the bridge surface and inside the bridge steel box girder.

Two Micaz nodes are used, node A sends packets and node B receives them. Node A sends 20 packets every second and transmits for 20 seconds. Node B returns ACK packet after it receives a packet from node A. We reduced the transmitting power for experiment inside the steel box girder. Every condition had been tested three times. The success rate is calculated by using the number of ACKs received divided by the number of packets sent.

TABLE 1. Communicatio Success Rate (%)

Transmitting RF Power	Distance						
	Bridge Surface			Steel Box Girder			
	30m	50m	100m	6m	12m	18m	36m
0dbm	89	87	82	97	86	50	41
	85	98	90	97	100	36	51
	93	90	52	89	60	20	77
-15dbm	-	-	-	78	70	96	52
	-	-	-	100	74	96	20
	-	-	-	76	81	88	96
-25dbm	-	-	-	69	78	34	0
	-	-	-	81	97	28	0
	-	-	-	-	87	88	0

Figure 2. The internal steel box grider

In Table 1, we can see that Micaz communication quality is very good within 50 meters on the bridge surface and still able to work at 100 meter distance. But communication in steel box girder is not very well; Figure 2 shows the inside condition, the whole structure is made by steel which is a natural wireless signal shield. The multipath propagation effect is shown in the internal of box girder. The radio environment is unstable; success rate varies from 60% to 100%. Experiment shows that reducing radio power can have better result in some cases, success rate shown in Table 1 increases when radio power decreases from 0dbm to -15dbm at the distance of 18 meters. This is a real challenge for design and deployment a WSN in the steel box girder. We have to make sure that every packet has been received and with no network crash.

B. System Architecture

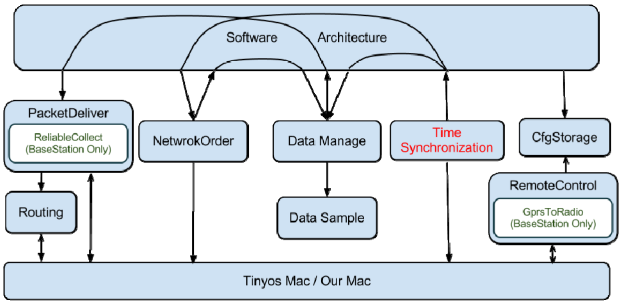


Figure 3. Embedded software top - level structure

Figure 3 describes the overall structure of the system. We decompose the system into separate components (see Table 2 for details), where each component can be improved without affecting the others. The "TimeSync" [17] component maintains globe wireless network time synchronization; this is used to make sure that every node can wake up at the beginning of every very time slices. "PacketDeliver" component delivers packet from one child node to parent node, this procedure continuous until packet has been delivered to the sink. Due to the poor environment in the box girder, in order to maintain the robustness of the wireless, the command packets are transmitted in broadcast. "NetworkOrder" component is used to make sure every node can get the message from the sink. We developed "CfgStorage" component to allow EEPROM to save the configuration data, so that node can keep the configurations after reboot. With the help of "RemoteControl" component, we can read and write the configurations through wireless commands. "Routing" component, similar to [18], maintains the routing table. "GprsToRadio" component can translate the commands received from GPRS and broadcasting commands through "RemoteControl" components. "ReliableCollect" component checks the packets from every node, and asks the node to resend if missed.

TABLE 2. EMBEDDED SOFTWARE COMPONENTS

Component	Functions
TimeSync	Maintain wireless network time synchronization
PacketDeliver	Deliver single packet to next hop
NetworkOrder	Broadcast network command
Routing	Self-organizing routing
CfgStorage	Save the configurations in non-volatile memory
RemoteControl	Reboot or change configurations by radio orders
GprsToRadio (Sink Only)	Relay message from GPRS to wireless network
ReliableCollect (Sink Only)	Check the packets from every node, ask to resend if missed

C. PacketDeliver Component

For a multi-hop network, not every node can direct connect to the sink. Therefore all nodes must have the ability to relay other nodes' packets. Each transmitted data packet requires an ACK confirmation due to the instability of the wireless communication. In Figure 4, after node 2 receives a packet from node 3, node 2 must send an ACK back.

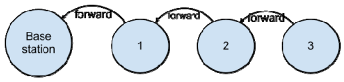


Figure 4. Multi-hop network diagram

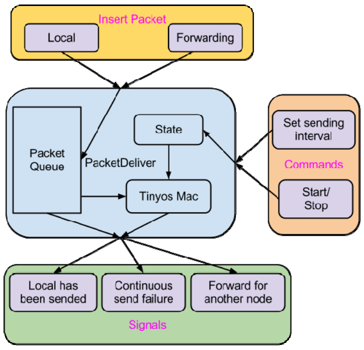


Figure 5. PacketDeliver component input/output diagram

In Figure 5, "PacketDeliver" component is focused on forwarding packets to the next hop. There are two types of source where packets can insert into forwarding queue: local generated data and data relayed from other nodes. It is necessary to control the transmitting rate to avoid network congestion. The "PacketDeliver" component can set the interval of continuous sending packet. After the local packet has been sent, the top-level (section B) receives a signal and can allow the next packet to be inserted to the queue. If the link quality is not good, the node keeps sending the packets inside the queue until it receive the ACK. If sending fails too many times, the top-level gets a signal for consider re-establish the route.

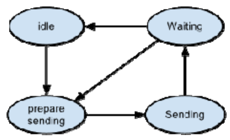


Figure 6. PacketDeliver state transition table

As shown in Figure 6, "PacketDeliver" has four states:

- Idle: forwarding queue is empty
- Prepare sending: get a packet from the head of queue
- Sending: sending the packet to next hop, and confirm the ACK packet to decide whether to resend
- Waiting: waiting for the time interval

D. Collecting Data at Time Slice

At the beginning of the time slice, every node wakes up and starts collecting data. As mentioned in the section I, sensors divided into two categories by sampling frequency. The two category types collect data at different time slices.

*First category: Every sen*sor node only sends one sampled data to sink. Sink maintains a list of all sensor nodes. If one node's packet is missed, sink asks it to resend the packet. If the packet cannot be successfully transmitted to the sink within three retransmission attempts, the sink considers this node's system is crashed and asks the network to turn off until the next time slice.

Second category: vibration data usually samples at 50Hz, and lasts for one minute. After sampling, each vibration node has hundreds of packets to be sent. In order to prevent collision, each time slice is only allowed for sending vibration data from one node.

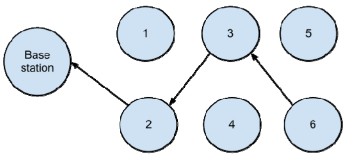


Figure 7. Collecting accel data

Figure 7 shows that when a vibration node is sending data, not everyone is needed. Therefore when the time slice is for vibration data, only the related nodes are working. Data are sent through the relay nodes in pipeline mode. When sink gets the final packet from the sending node, it checks if all packets have been received. If some were lost, sink send NACK to the node ask it to resend the missing packets.

Comparing with CTP, the designed system protocol can control the node's RF chip. Other unrelated node interference can be turned off for vibration data transmission. In this case, the data throughput can reach very high speed.

IV. Evaluation at Lab

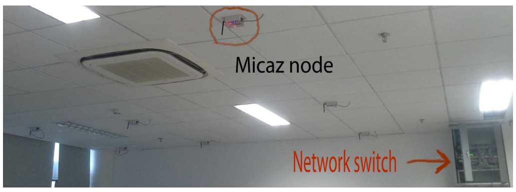


Figure 8. The actual picture of the test bed, inside the red circle is a Micaz node, and a network swith is used to conneting nodes to a server

As shown in Figure 8, a test bed was installed in the laboratory celling. The node is a Micaz with Ethernet ability. There are total of sixteen nodes installed in the room, they are connected by the Ethernet cable to the server. The server allows the reprogram and control of the test bed nodes.

The RF transmitting power is reduced for the tests in test bed. One node plays the part of sink node and 4 nodes send vibration data, 4 nodes sends temperature data, and 6 nodes have a role of relay. The remaining node is used as a sniffer for monitoring the entire network.

The test program is tested for over a week with no problem shown.

V. Deployment

Figure 9. Several types of sensor nodes deployed in the field

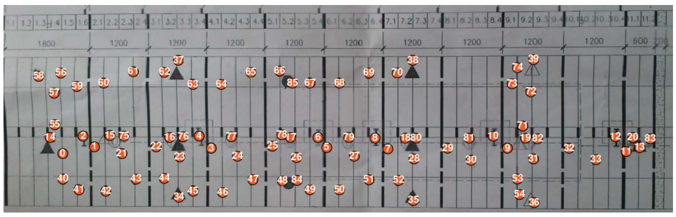
Two networks are deployed in the field on Ronghu Bridge, one for the bridge surface, and another is in the steel box girder.

The sink nodes are connected to GPRS (General Packet Radio service) nodes with service provided by China Mobile. Though the GPRS, we can receive packets and send commands to the wireless network.

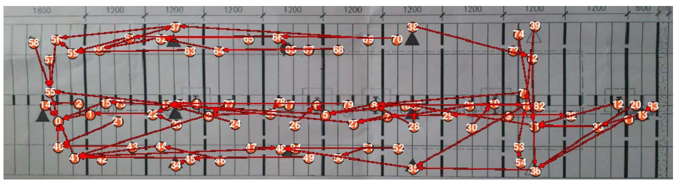
There are total of 164 sensor nodes installed on the bridge. The largest sub network contains 86 nodes. Most nodes are deployed on the bridge surface and inside the steel box girder.

On the bridge surface, sensors deployed are: 1 sink node, 12 cable tension nodes, 3 bridge tower tilt nodes, 1 bridge tower vibration node, 4 bridge desk vibration nodes, 1 wind direction/speed node, and 5 relay nodes. All of these nodes have solar panels and can be charged during the day time. Solar panels allow these nodes to operate for at least several years.

Inside the steel box girder, we deployed 1 sink node, 13 vibration nodes, 13 strain nodes, 11 bridge deck settlement nodes, and 48 relay nodes, as shown in Figure 10a. These nodes use lithium batteries for power supply. The battery contains 4 cells with 76Ah in total. As described above, the harsh environment results a very complex routing table (Figure 10b). Therefore, nodes' communication quality is not relative to the physical distance.



(a)



(b)

Figure 10. Nodes in the steel box girder, the photo above just identify a total of 85 nodes, the following has a routing topology

The time slice is set to last every two minutes, with work time and sleep time varies according to the network condition. The first category data (cable tension, bridge tower tilt, wind, steel box girder strain, and bridge deck settlement) can be collected within 10 seconds, the node goes to sleep for the rest of 110 seconds. For the vibration data, we sample at 50Hz, sampling lasts from 81.92 or 40.96 seconds (the sampling time could be changed through GPRS), so one node should send 128 or 256 packets (one packet contains 32 Byte of vibration data) after sample. Due to the poor environment in the steel box girder, the vibration data are relayed through a very unstable route. Also different vibration nodes are located at different route depth. Early deployment shows that vibration data can finish transmission between 15 to 50 seconds. Vibration data only samples 4 times a day to save battery power. In such working conditions, we expect the battery can work for at least 1 year.

The deployment started from June 19, 2012, when 12 cable tension nodes installed on the bridge surface. The 11 bridge deck settlement nodes in the box girder are installed last on July 30, 2012. The deployed WSN is working stably until now. We keep all the received data stored in the database. At September 20, 2012, the total data received is as follows:

- cable tension: 53059 packets, 2971 KB
- wind: 48171 packets, 1011 KB
- bridge tower vibration: 312626 packets, 14068 KB
- bridge tower tilt:91755 packets, 2569KB
- steel box girder strain: 643044 packets, 19291 KB
- steel box girder vibration : 486618 packets, 21898 KB
- bridge deck settlement (including temperature message): 378133 packets, 8319 KB

VI. CONCLUSIONS

A wireless sensor network on Ronghu Bridge with multiple sensors was designed and deployed. This system is expected to work for at least five years and has been working well for three months.

There are two major contributions of this application. The first is that this system is able to collect multiple types of sensor data where most data can be transmitted within 10 seconds and vibration data can be transmitted between 15 to 50 seconds. Second, this system has collected over 70 MB of sensor data, which is able to have a good analysis of the structure of the Ronghu Bridge. These data are very valuable to the SHM research area. We also learned valuable lesson for field deployment.

ACKNOWLEDGMENT

This work was supported in part by National Science and Technology Major Project, 2010ZX03006-003-01 and by a grant from Wuxi Government.

REFERENCES

[1] S Kim, “Wireless Sensor Networks for High Fidelity Sampling,” P. thesis, Univ. of California Berkeley, California, Unite States, 2007.

[2] B. F. Spencer Jr., and C. B. Yun, “Wireless Sensor Advances and Applications for Civil Infrastructure Monitoring,” University of Illinois at Urbana-Champaign, NSEL Report No. 024. 2010.

[3] T. I. Miller, B. F. Spencer Jr, J. Li, and H. Jo, “Solar Energy Harvesting and Software Enhancements for Autonomous Wireless Smart Sensor Networks,” University of Illinois at Urbana-Champaign, NSEL Report No. 022, 2010.

[4] S. Cho, S. A. Jang, H. Jo, K. Mechitov, J. A. Rice, H. Jung, C. Yun, B. F. Spencer Jr., T. Nagayama, and J. Seo, “Structural health monitoring system of a cable-stayed bridge using a dense array of scalable smart sensor network,” in *Proc. SPIE’10,* 2010, vol 7647.

[5] W. Liu, X. Fei, T. Tang, P. Wang, H. Luo, B. Deng, and H. Yang, “Application Specific Sensor Node Architecture Optimization—Experiences from Field Deployments,” in *Proc. ASP-DAC’12,* 2012, pp389-394.

[6] MEMSIC MicaZ Mote. [Online]. Available: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

[7] TinyOS home page. [Online]. Available: http://www.tinyos.net/.

[8] CC2420 Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee™ Ready RF Transceiver. [Online]. Available: http://www.ti.com/product/cc2420.

[9] C. K. Rupani, and Dr. T. C. Aseri, “Comparative Study of Transport Control Protocols for Wireless Sensor Networks,” in *Proc. ICISA’10,* 6 February 2010, pp317-321.

[10] S. N. Pakzad, G. L. Fenves, S. Kim, and D. E. Culler, “Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring,” in *Journal of Infrastructure Systems,* Vol. 14, No. 1, pp89-101, March 1,2008.

[11] S. Kim, S. N. Pakzad, D. Culler, J. Demmel, G. L. Fenves, S. Glaser, and M. Turon, “Health monitoring of civil infrastructures using wireless sensor networks,” in *Proc. IPSN’07,* 2007, pp254-263.

[12] J. Paek, and R. Govindan, “RCRT: Rate-Controlled Reliable Transport Protocol for Wireless Sensor Networks,” in *ACM Transaction on Sensor Networks,* Vol 7, No. 3, Article 20, 2010 .

[13] R. Musaloiu-E, C. Liang and A. Terzis, “Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks,” in *Proc. IPSN’08*, 2008, pp421-432.

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol", in *Proc. SenSys’09*, 2009, pp1-14.

[15] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale, “PIP: A Connection-Oriented, Multi-Hop, Multi-Channel TDMA-based MAC for High Throughput Bulk Transfer,” in *Proc. SenSys’10,* November 3–5, 2010, pp15-28.

[16] S. He, J. Chen, D.Y.Yau, and Y. Sun, “Cross-layer Optimization of Correlated Data Gathering in Wireless Sensor Networks,” in *Proc. SECON’10,* 2010, pp 1-9.

[17] M. Maróti, B. Kusy, G. Simon and A. Ledeczi. “The Flooding Time Synchronization Protocol,” in *Proc. SenSys’04,* November 3–5, 2004, pp39-49.

[18] Philip Levis, Neil Patel, David Culler and Scott Shenker. "Trickle: A Self-Regulating Algorithm for Code Maintenance and Propagation in Wireless Sensor Networks." In *Proc. NSDI’04*, 2004, pp2.