Research Article
Real-Time Seismic Data Acquisition via a Paired Ripple Transmission Protocol

Jin-Ling Lin, 1 Kao-Shing Hwang, 2 Hui-Kai Su, 3 and Min-Che Hsieh 4

1 Shih Hsin University, Taipei 11678, Taiwan
2 National Sun Yat-sen University, Kaohsiung 80424, Taiwan
3 National Formosa University, Yunlin 63201, Taiwan
4 National Chung Cheng, Chiayi 62102, Taiwan

Correspondence should be addressed to Jin-Ling Lin; jllin@cc.shu.edu.tw

Received 24 November 2012; Revised 4 April 2013; Accepted 23 April 2013

Copyright © 2013 Jin-Ling Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work uses a low-cost, reliable, and microchip-based wireless transmission solution to real-time collect earthquake data across local and wide areas. A transmission chain consisting of sensor units (nodes), each transmitting earthquake data unidirectionally to the end, is proposed. Each node consists of a seismic sensor, analog digital converter, radio frequency module, and a microchip for central control. The terminal node is responsible for transmitting data to a display server, which collects and analyzes all earthquake data from different transmission chains. Moreover, users also can distribute nodes, plug-in computers, in a wide area to monitor earthquake activities and transmit data to a web server. Then interested people can view the circumstance of an earthquake via web maps. For efficient wireless transmissions and to maximize bandwidth usage, a modified ripple protocol is applied to the wireless transmission between nodes in a daisy chain. Field experiments verify the practicality of the proposed system.

1. Introduction

Taiwan is located in an active earthquake zone on the western Pacific Ocean Rim. The convergence of the western Eurasian plate and the eastern Philippine Sea plate causes Taiwan's earthquakes. The danger to property and humans from earthquakes is serious. Thus, developing a seismic monitoring system, which can catch an earthquake's signals and analyze earthquake data is important [1–5].

Sensors are often used in harsh environments. Therefore, how to develop a reliable sensing solution for monitoring the variation of environment becomes very challenging [6]. A seismic monitoring system needs many sensors, sited at many locations, and all of these sensors must send signals to a collection center. The system consists of sensors, signal converters, data transmitters, data storage, and a user interface. First, sensors create signals using analog voltage. Next, signal converters change the analog signals to digital signals and then transmit the digital data to a data storage device. Finally, users can access the data and analyze it via the user interface. Data acquisition is the basic unit for the success of monitoring system [7]. An overview of a seismic monitoring system is shown in Figure 1.

When the system uses wired transmission, all sensors connect directly to the signal converter so the system is very reliable and can transmit a lot of data simultaneously. However, it is very inconvenient and time consuming to set up a wired transmission system. Wireless transmission overcomes the drawbacks of wired transmission [3–5, 9]. Besides, it is more practical as the system can observe very small signal changes with high resolution. So it would be better if this system could convert analog signals to high-resolution digital signals. A practical and reliable seismic monitoring system should be designed according to these requirements: wireless transmission, high-resolution analog-digital conversion, good reliability, convenience of accessing data, and low cost. Ripple transmission is reliable than the regular wireless transmission in such a seismic monitoring system, but it works on a computer instead of a microchip. Therefore, the article proposed the design of a cheaper RF
module with a secure transmission protocol which is actually a simplified version evolving from the well-known ripple protocol.

The paper is organized as follows. Section 2 discusses background research into ripple wireless transmission protocol. Section 3 presents the architecture of the proposed seismic monitoring system. Section 4 discusses the research approach and system implementation. Section 5 presents the experiments and a discussion. Conclusions are provided in Section 6.

2. Ripple Wireless Transmission Protocol

For wireless transmission, the most popular protocol [10] is carrier sense multiple access with collision avoidance (CSMA/CA). CSMA/CA is a protocol used to prevent data collisions. Once a node wishes to transmit data, it has to first listen to the channel for a predetermined amount of time to determine whether or not the channel is clear within the wireless range. If the channel is clear then the node is permitted to begin transmission; otherwise the node defers its transmission for a random period of machine cycle time. The use of collision avoidance is used to improve the performance of CSMA by attempting to divide the wireless channel up somewhat equally among all transmitting nodes within the collision domain. One of the problems of wireless data communications with CSMA/CA protocol is the hidden node problem. Figure 2 shows the hidden node problem, whereby node S1, in range of the receiver R, is not in the range of node S2; therefore, there is no way for S1 to know whether S2 is transmitting to R.

IEEE 802.11 RTS/CTS exchange supplements CSMA/CA by adding the exchange of a request to send (RTS) packet sent by the sender S and a clear to send (CTS) packet sent by the intended receiver R, such that all nodes within the range of the sender, receiver, or both are alerted not to transmit for the duration of the main transmission. The use of collision avoidance is used to improve the performance of CSMA by attempting to divide the wireless channel up somewhat equally among all transmitting nodes within the collision domain. One of the problems of wireless data communications with CSMA/CA protocol is the hidden node problem. Figure 2 shows the hidden node problem, whereby node S1, in range of the receiver R, is not in the range of node S2; therefore, there is no way for S1 to know whether S2 is transmitting to R.

IEEE 802.11 RTS/CTS exchange supplements CSMA/CA by adding the exchange of a request to send (RTS) packet sent by the sender S and a clear to send (CTS) packet sent by the intended receiver R, such that all nodes within the range of the sender, receiver, or both are alerted not to transmit for the duration of the main transmission. Thus, the hidden node problem can be solved [11]. But, CSMA/CA with RTS/CTS exchanges cause another transmission problem—the exposed node problem, which occurs when a node is prevented from sending packets to other nodes due to a neighboring transmitter [12]. Suppose node S1 and node S2 want to send packages to node R1 and node R2, respectively, as shown in Figure 3. When node S1 sends an RTS packet to node R1, node R1 will send a CTS packet to node S1 and node S2, because S1 and S2 are in the transmission range of R1. In such a case, node S2 cannot send its data package to node R2 while node S1 is sending its data package to node R1. It is obvious that S2 can still send its data packages to R2 without interfering with the transmission between S1 and R1. The channel between S2 and R2 is wasted in such a case. Thus, a ripple protocol for wireless transmission is proposed to solve these transmission problems, within the CSMA/CA protocol.

Some wireless transmission protocols in the fashion of rippling [13, 14] is proposed to solve hidden node and exposed node problems, which arise when the CSMA/CA protocol is applied. In general, a ripple protocol with a chain topology uses a decentralized controlled-access approach to protect nodes from unintentional packet collisions, so it has the advantage of an ability to reuse space when compared with CSMA/CA protocol. Nodes are equally spaced, and radio nodes that are not neighbors do not interfere with each other in a ripple protocol. The ripple protocol has the advantages of identical ranges both in interference and transmission and a fixed elapsed time in data packet transmission.

A ripple protocol is proposed in this article which, as the protocol in [8], uses six types of frames: DATA, NULL, ready-to-send (RTS), clear-to-send (CTS), acknowledge (ACK), and ready-to-receive (RTR). The RTS and RTR frames are treated as “tokens,” which is the same as other token-passing protocols. A node, in a ripple protocol, is allowed to send a DATA frame only if it holds a token. The way to generate
Exposed node problem

Currently transmitting

Wants to transmit

Broadcast ranges of each node

Figure 3: Exposed node problem.

and circulate RTS and RTR tokens depends on the state of the node. As shown in Figure 4, each node has four states.

(i) Transmit (TX): a node will enter this state if it is ready to send a DATA frame.

(ii) Receive (RX): a node will enter this state if it is ready to receive a DATA frame.

(iii) Listen: a node will enter this state if it overhears either CTS frame (for a hidden node) or RTS frame (for an exposed node). A node in Listen state must keep silent waiting for a network-allocation-vector (NAV) and may transmit an RTR token to its upstream node if the channel is sensed to be clear for IFS$_{RTR}$ (the interframe space of an RTR frame) after the expiry of NAV.

(iv) Idle: it is the initial state for all nodes. A node will enter this state if it has been interrupted by unexpected conditions when in the TX, RX, or Listen state.

IFS$_{RTR}$, the interframe space (IFS) of an RTR frame, is set as two SIFS plus the time needed to transmit a RTS frame and defined as follows:

$$\text{IFS}_{\text{RTR}} = \text{SIFS} + T_{\text{RTS}} + \text{SIFS} = 2\text{IFS}_{\text{RTS}} + T_{\text{RTS}},$$

where the IFSs of the remaining frames are all set as short IFS, SIFS. In the proposed system, IFS$_{\text{RTR}}$ is treated as a one-time unit.

3. Architecture of the Seismic Monitoring System

The data type for wireless transmission must be digital, so seismic sensors, combined with analog digital converters (ADC) and wireless transmission, are the basic unit for the proposed seismic monitoring system, and the basic unit is treated as a node in the system. These nodes combined with a terminal node, used to collect data, form a transmission chain, as shown in Figure 5.

Each node in the transmission chain can receive analog signals, convert analog signals to digital signals, and transmit these digital signals to the next node. Each node not only receives digital data from its previous (upstream) node in the transmission chain but also transmits the data, including the data collected from itself and its previous node, to the next (downstream) node. The terminal node connects or is embedded into a computer, which analyzes the collected earthquake data and transmits it to a remote web server for further processing.

Considering the features of cost [3], data rate, and development difficulty, RF is preferred as the wireless transmission device module, instead of Bluetooth, Wi-Fi, or ZigBee. For the high-resolution digital signal, LTC2440 [15], which is a 24-bit high-speed differential delta sigma ADC with selectable speed/resolution and with an output rate up to 3.5 KHz, was chosen as the ADC converter. Finally, the microcontroller dsPIC33FJ128MC802 [16] was selected as the central controller of each node, instead of 8051 or ARM, when cost, signal processing ability, and development difficulty were taken into consideration.

4. System Implementation

One of the major objectives of this work is to alleviate the burden of carrying a bunch of communication cables to connect sensory nodes in deployment. IEEE 802.11 solution may be a good choice to resolve this concern. Whereas, while the cost and setup should be taken into considerations, this solution may turn out to be unfeasible, especially dealing with the tremendous number of deployed nodes. Therefore, the article proposed the design of a cheaper RF module with a secure transmission protocol which is actually a simplified version evolving from a well-known ripple protocol.

The integrity of data packages and transmission efficiency is the most important issue in wireless transmission. To maximize the usage of wireless transmission bandwidth and preventing packet collisions, the ripple protocol is ported on dsPIC8022 microchip and modified as a paired ripple transmission protocol (PRTP) such that it can appropriately work in the proposed system.

The paired ripple transmission (PRT) works on a microchip (dsPIC802s) instead of a computer, which the ripple transmission usually works on. The dedicated device mainly consisted of a 16 bits dsPic but capable of a reliable RF data transmission so as to being dispersed broadly in a field. The handshaking packet format of ripple transmission protocol is simplified and modified to 3 items: TYPE, NODE, and DES_NODE, and each item accounts for 1 byte only. TYPE stands for the type of packet, for example, RTS, CTS, RTR, or ACK. NODE and DES_NODE describe the source and destination of packet, respectively. Because the length of a DATA package is fixed, the network allocation vector (NAV) for each transmission is the same. Each node has to transmit on two different wireless channels, so the proposed paired ripple transmission needs two radio frequency (RF) modules—one is a receiving channel and the other is a sending channel. A node will communicate with its upstream node while the receiving channel and downstream node with the sending channel. The functions and processing flow of each node, which is ported with the proposed paired ripple transmission protocol, are described in Figure 6. Blue solid lines denote the possible input, and green dashed lines denote the output.
Receive RTR or CTS, and wait for SIFS
TX
Overhear RTS, CTS, or RTR
Listen
Receive data
CTS or RTR overhearing timeout
Idle
Receive data
timeout
CTS or RTR overhearing timeout
RX
Receive data
Overhear RTS, CTS, or RTR
Transmit data
CTS or RTR overhearing timeout

Figure 4: State diagram of a node in ripple protocol [8].

Figure 5: Structure of transmission in a daisy chain.

Figure 7 describes how a node changes its states with the paired ripple transmission protocol. Each node has two RF modules, one for receiving data packages from the upstream node, denoted by RX-RF, the other one for sending data packages to the downstream node, denoted by TX-RF. The background to the upper and lower part of an arrow represents the module RX-RF and TX-RF, respectively. When a node processes a handshake with other nodes, both of its two RF modules will enter TX mode to send packets to the upstream and downstream nodes. When a node listens to the channel or expects for a packet, both of its two RF modules are in RX mode. When a node is transmitting DATA frame, its RX-RF module will enter RX mode and the RF-TX module will enter TX mode.

The hardware needed in the proposed paired ripple transmission consists of a printed circuit board (PCB) and some electrical parts, shown in Figure 8. The dsPIC microcontroller takes charge of analog-digital conversion and RF transmission. Input data comes from the G-sensor and RF-RX. RS-232 and RF-TX are for output. Two RF modules are responsible for wireless transmission. If the node is the last one in the chain of transmission, data will be transmitted by RS-232 instead of RF. The power supply has two types of power sources: a 9 V Li-ion battery and a 5 V DC power jack. The 5 V power is supplied by a 5 V DC power jack or the output of a L7805, which is a positive voltage regulator and translates the voltage provided by the Li-ion battery from 9 V to 5 V. The 5 V power supply is used to drive a LTC2400,
Analog signal from sensors → Collect data → A/D convert → Add time stamp → Data package → RF receive → Transmit to downstream node → RF transmit → Data package to downstream node → Transmit to computer → Setup-info. package → Setup node number and time synchronization

**Figure 6:** Functions and processing flow of each node in chained paired ripple transmission.

**Figure 7:** State diagram of RF modules in the proposed paired ripple transmission.
G-sensor, and HIN232. The LM3940, in the power supply circuit, will reduce the 5 V down to 3.3 V and supply the dsPICs and RF modules.

Remote earthquake data acquisition and analysis need software tools to help with data acquisition, transmission, collection, and analysis. The software tools needed in the transmitting nodes, data-collection computer, and data-analysis server are described as following.

1. Software Tools for the Transmitting Node. The software architecture in a node is divided into several parts: initial functions, ADC data framing functions, and RF communication functions, as shown in Figure 9. Initial functions are responsible for all of the hardware device initialization such as I/O mapping, communication interface, timer, and RF module initialization. The ADC data framing function is responsible for getting ADC value and combining it with a time stamp to be framed as an ADC data package. RF communication functions process data packages both from RF for RX and its own pair dsPIC802 for ADC. Then they transmit the input data package to the downstream node or computer.

2. Software Architecture for the Data Collection Computer. The main mission of the data collection computer is transmitting data from the end node to the web server and doing time synchronization with the nodes. Users can set up the ID of each node and send time synchronization information at the same time. If the node is combined with the data collection computer, its ID is localhost, whose IP is 127.0.0.1; otherwise the ID can be any effective IP address. The data collection computer listens to COM port and receives messages continuously. Any message coming from COM port will be transmitted to the web server, by socket, immediately. In addition, this program can send a package which includes information about node setup and system time on the computer. Also, a visual component library (VCL), called TYbCommDevice, was added such that the program can support all COM ports.

3. Software Architecture for the Data Analysis Server. The main purpose of the server software is to make the APM work stably and build a friendly user interface. The program on the server can receive HTTP POST data packages from the socket and store data after analyzing the data package. The program can also respond to users’ requests on the web browser, as shown Figure 10.

5. Experiments and Discussion

The dedicated device mainly consisted of a 16 bits dsPic but capable of a reliable RF data transmission so as to being dispersed broadly in a field. Therefore, to evaluate the performance of the proposed paired ripple transmission (PRT), the PRT was compared with the regular wireless transmission (RWT)—RF without using ripple transmission. The baud rate of the RS-232 is 115200, the length of each data package was 17 bytes, and the wireless transmission speed was 1 Mbps in the experiments. The numbers of data packages, transmitted and recorded on the server, were accounted for each minute.

First, the maximum efficacy of nodes in ripple protocol was tested. In Figure 11, the effective data, represented by a red line with square marker, means corrected data is recorded at the web server, the timeout missing data, represented by a
green line with triangle marker, means these are remaining packages, which have not been sent to the web server, and the error data, represented by a black line with star marker, means the packages were garbled or in the wrong format. In Figure 11, someone can easily observe that the effective data is less than invalid data only when each node takes 125 samplings per second (SPS), since higher SPS would make the system too busy to process data. Besides, the error data increases when SPS becomes higher. In other words, the seismic monitoring system could work well when SPS is less than 125.

Figures 12 and 13 show the status of data packages for the proposed paired ripple transmission, which has 6 nodes in the chain of ripple transmissions and the regular wireless transmission, respectively. The brown brick denotes effective data, sky blue solid rectangles denote error data, and dotted
gray rectangles denote timeouts-missing data. Since the regular transmission could result in the overrun of memory buffers and system crashes, the efficiency of transmission is affected. Ripple ensures an orderly transmission, but it still misses a lot of data when the data flow is overloading the transmission chain. From the experimental results, it is obvious that higher SPS leads to the less effective data both in paired ripple and regular transmissions. The regular transmission has much more error data than the proposed paired ripple transmission, as shown in Figures 12 and 13, and the number of effective data packages in paired ripple transmission is more than in regular transmissions, as shown in Figure 14, where the solid line represents the amount of effective data for PRT and the dashed line represents the amount of effective data for RWT.

6. Conclusions

A paired ripple transmission protocol was proposed and implemented into modular hardware. The ripple protocol can improve transmission efficiency by spatial reuse in real-world applications. When the transmission chain gets overloaded, the regular wireless transmission has errors in large data packages, but the ripple transmission maintains the integrity of data packages. Because error data brings extra payloads to nodes, it makes the system crash. Thus, the proposed paired ripple transmission is more efficient than the wireless transmission without ripple. A wide area deployment is pictured, making the function of the designed work pieces more pervasive. A web server was set up for easy data representation, and a lot of cheap hardware parts were chosen, such that each node costs less than 30 dollars.

References

[1] Y. Pang and X. Liu, “The design of the intelligent and fieldbus sensor for measuring the level and temperature of the water well monitoring earthquake,” in Proceedings of the 41st SICE Annual Conference (SICE ’02), pp. 569–574, August 2002.
[2] T. C. Hutchinson and F. Kuester, “Monitoring global earthquake-induced demands using vision-based sensors,” IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 1, pp. 31–36, 2004.

[3] T. Takano and T. Maeda, “Experiment and theoretical study of earthquake detection capability by means of microwave passive sensors on a satellite,” IEEE Geoscience and Remote Sensing Letters, vol. 6, no. 1, pp. 107–111, 2009.

[4] R. Tan, G. Xing, J. Chen, W. Z. Song, and R. Huang, “Quality-driven volcanic earthquake detection using wireless sensor networks,” in Proceedings of the 31st IEEE Real-Time Systems Symposium (RTSS ’10), pp. 271–280, San Diego, Calif, USA, December 2010.

[5] J. Nachtigall and J. P. Redlich, “Wireless alarming and routing protocol for Earthquake Early Warning Systems,” in Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS ’11), Paris, France, February 2011.

[6] A. Ferreira Da Silva, A. F. Goncalves, L. A. De Almeida Ferreira et al., “A smart skin PVC foil based on FBG sensors for monitoring strain and temperature,” IEEE Transactions on Industrial Electronics, vol. 58, no. 7, pp. 2728–2735, 2011.

[7] P. Mariño, F. Poza, S. Otero, and M. A. Domínguez, “Reconfigurable industrial sensors for remote condition monitoring and modeling,” IEEE Transactions on Industrial Electronics, vol. 57, no. 12, pp. 4199–4208, 2010.

[8] R. G. Cheng, C. Y. Wang, and J. S. Yang, “Distributed medium access protocol for wireless mesh networks,” U.S. Patent 7822009, http://www.freepatentsonline.com/7822009.html.

[9] R. C. Luo and O. Chen, “Mobile sensor node deployment and asynchronous power management for wireless sensor networks,” IEEE Transactions on Industrial Electronics, vol. 59, no. 5, pp. 2377–2385, 2012.

[10] H. Liu, B. Zhang, H. T. Mouftah, X. Shen, and J. Ma, “Opportunistic routing for wireless ad hoc and sensor networks: present and future directions,” IEEE Communications Magazine, vol. 47, no. 12, pp. 103–109, 2009.

[11] H. Jasani and N. Alaraje, “Evaluating the performance of IEEE 802.11 network using RTS/CTS mechanism,” in Proceedings of the IEEE International Conference on Electro/Information Technology (EIT ’07), pp. 616–621, Chicago, Ill, USA, May 2007.

[12] J. Jafarian and K. A. Hamdi, “Analysis of the exposed node problem in IEEE 802.11 wireless networks,” in Proceedings of the 11th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, 2010.

[13] R. G. Cheng, C. Y. Wang, L. H. Liao, and J. S. Yang, “Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks,” IEEE Communications Letters, vol. 10, no. 2, pp. 123–125, 2006.

[14] R. G. Cheng, C. Y. Wang, and L. H. Liao, “Ripple: a distributed medium access protocol for multi-hop wireless mesh networks,” in Proceedings of the IEEE 63rd Vehicular Technology Conference (VTC ’06), pp. 289–293, Melbourne VIC, Australia, July 2006.

[15] Linear Technology Inc, “LTC2440—24-bit high speed differential delta sigma ADC with selectable speed/resolution,” 2012, http://www.linear.com/product/LTC2440.

[16] Microchip Technology Inc., “16-bit dsPIC digital signal controllers—dsPIC33FJ128MC802,” 2012, http://www.microchip.com/wwwproducts/devices.aspx?DocName=en532302.
Submit your manuscripts at
http://www.hindawi.com