A Hybrid, Dynamic Traffic-Adaptive MAC Protocol for Wireless Sensor Networks

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

*Department of Electronic Engineering, Tsinghua University, Beijing, China

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

Abstract—**For wireless sensor network applications, such as Structural Health Monitoring (SHM), data traffic adaptive becomes more and more important for Medium Access Control (MAC) protocol design. In this work, we present a new hybrid MAC protocol. This protocol is able to adapt for both low and high traffic load. For better energy efficiency and network scalability, we use improved Carrier Sense Multiple Access (CSMA) for handling low data traffic where hidden node problem is resolved for better performance under interference. When high traffic load is transmitted in the network, to achieve a low data retransmission rate and low interference, a receiver initiated burst transmission protocol is implemented for continuous multiple data relay. Minimum overhead is used in this burst traffic to control the flow rate as well as to resolve the interference. Hop by hop data retransmission is only used for missing segment of the burst data. This protocol has been evaluated in an indoor testbed, and it is now being tested in a real SHM application in Ronghu Bridge, Wuxi, Jiangsu province, China. It is robust against interference and achieves over 16% lower data retransmission rate and over 38% lower energy consumption, especially in burst traffic, comparing to TinyOS supported CSMA protocol.**

Keywords—**Wireless Sensor Networks, MAC protocol, Hybrid, Traffic Adaptive, Burst Transmission, Structural Health Monitoring**

I. INTRODUCTION

Research on Wireless Sensor Networks (WSNs) has gain massive growth in recent years. Areas from protocol design to field applications have achieved many successes and showed good promises for the future. However, these research areas are still in early stages. For example, MAC protocols used for WSNs are not perfect: they were usually developed to solve one or few problems, such as data throughput, energy efficiency, or network scalability [1]. Also, most WSN applications are still experimental. It is far away from massive production and implementation.

Structure Health Monitoring (SHM) is one of the areas that can benefit from the WSN technology. Real world applications have been installed and tested. Among those, several studies for bridge SHM have shown good results [2]-[6]. Traditional SHM for bridges uses wired sensor for gathering data. It requires large amount of installation work and periodic inspections. Using wireless communication can be a good solution, where installation cost can be reduced to the minimum and maintenance period can be extended to once every few years. In order to evaluate the health of a typical cable stayed bridge, sensors including accelerometers for vibration and cable tension, strain sensors for steel stress, displacement sensors for settlement are installed in [6]. Different sensors have different sampling frequency. Most sensors are sampled every few minutes individually, but vibration sensors are sampled together. As vibration data need to be used for spectral analysis, they have to be sampled in very large qualities.

The MAC protocol used in [6] is provided by TinyOS 2.x environment, which is a simple CSMA/CA based protocol with low power listening ability. This protocol works fine for distributed WSNs under the condition that the throughput is low. However, it is not suitable for high traffic solutions. TDMA based protocols can achieve much higher throughput [1], but these protocols are not suitable for large scale applications.

The proposed MAC protocol provides a hybrid solution by combining CSMA/CA with a receiver initiated burst transmission when high traffic is needed. The CSMA/CA is improved to avoid hidden node problem, which also has better performance under interference. The receiver initiated burst transmission uses very few control frames to allow nodes to transfer large amount of data with minimum data lost.

The rest of this paper is organized as follows. Related work in MAC protocols is shown in section II. The proposed hybrid protocol is described in section III. Testbed implementation and evaluation are described in section IV. We conclude this paper in section V.

II. RELATED WORK

A typical WSN application usually involves deploying tens to hundreds of nodes in a large area. Battery is usually the main power source and it is required to last up to a few years. One of the main design concerns is how to extend the lifetime of the network. For a typical MAC protocol, there are several sources that can deplete battery energy fast [1]:

- Collision: multiple messages transmitting at the same time resulting a collision and waste energy for both transmitters and receivers.
- Overhear: unrelated or redundant messages are received.
- Overhead: protocol uses overhead such as Request to Send/Clear to Send (RTS/CTS) packets to avoid

collision. They drain battery energy but carry no any useful information.

- Idle Listening: node has radio on but not transmitting or receiving anything, battery energy is also wasted during this time.

IEEE 802.11 [7] uses RTS and CTS packets to reserve the channel before sending the actual message. Neighbour nodes who receive the RTS or CTS packet and not participating in the transmission go to sleep. This solves the hidden node problem for potential data collision and saves battery energy by putting other nodes into sleep. But these overheads are considered inefficient for the communication, which is only recommended to transfer longer packets.

SMAC protocol [8] divides time into active and sleep period. Nodes work on active time to exchange data. Synchronization is done in the beginning of an active period, followed by data packets. The idle listening is reduced by putting nodes into sleep when they are not working. The protocol also tends to reduce collision by sending the RTS/CTS overheads before transmitting the packets. It also fragments the long packet into small packets and transmits them in burst. RTS/CTS could be used to reserve the channel for the entire burst. The main reason for burst transmission is to reduce the cost of packet retransmission, because small fragment packet retransmission consumes less energy than one long packet. This protocol performs well in slow to medium traffic networks. However, predefined active period may not be suitable for all the conditions, nodes who wakes up but do not need to transmit or receive packet end up wasting their energy.

To further reduce energy, nodes are put into sleep for most of their time and wake up only when they are needed. Low Power Listening is one approach. Node who needs to send a packet should first send a long enough preamble that wakes up the receiver. Receiver periodic wakes up to detect the channel for preamble, it goes back to sleep if the channel is clear, or stays awake for the incoming packet. CCA (Clear Channel Assessment) is a technology used to detect the channel status, but sometime can result in false positive because the channel is noisy. BMAC [9] improves CCA by using outlier detection. This technique allows the node to better determine the channel status. The outliers can be detected in the received signal if the channel energy is much below the noise floor. The noise floor can be averaged with signal strengths sampled when channel is clear. This protocol is supported by TinyOS 2.x environment. It performs well when throughput is very low, where low power listening can save battery energy. To achieve higher throughput, it uses the CSMA/CA protocol only without low power listening mechanism. But the throughput is still limited, which could not be suitable for high traffic networks.

Protocol improved from BMAC, such as X-MAC [10] considers that long preamble is an energy wasting source. The protocol cuts the long preamble into several short preambles with time gap in between. Receiver who wakes up early can reply with an ACK and starts the packet transmission early. This will save the transmitter's battery energy and the operation time for both nodes. Low Power Probing is a similar protocol described in [11] but in reverse. Receiver probes the channel periodically asking for packet to send. Transmitter that has data to send needs to listen to the channel first for receiver's probe, and then transmits the data. Since periodic probing needs to consume energy even there is no sender, it is less energy efficiency than low power listening. But study in [12] shows that under interference, low power probing has better throughput than X-MAC, and receiver's power consumption is lower too.

For more complicated networks, hybrid protocols are usually used. Hybrid protocols combine two or more simple protocols together to achieve better performance under different traffic variation. IEEE 802.15.4 [13] uses a very flexible MAC. CSMA/CA protocol is used for non-beacon mode, where network is usually simple and performs only simple operations. Beacon mode is used for more complex network. The protocol uses beacon between each time frame. Inside a time frame, nodes transmit data in the active period. Data can only be sent outside the reserved time slot defined by the beacon. If no reservation is done, nodes uses CSMA/CA inside contention access period to compete the channel before transmit data. Although this protocol performs well, the complied stack is large and cannot be installed into hardware limited nodes.

ZMAC uses CSMA for low traffic condition and switches to TDMA when high traffic load is needed in the network [14]. Synchronization for TDMA is done by distributed slot allocation algorithm which clears the hidden node problem in the two-hop neighbourhood. However, schedule drift requires the protocol to re-run the synchronization periodically thus reduces the battery energy.

Another interesting study in [15] shows that, the energy consumption between data push and pull are different. For a typical push protocol, the maximum and average power consumption stays constant. However, for a typical pull protocol, the longer the pull interval the lower the maximum and average power consumption. Therefore, if the pull interval is long enough, it can save more energy than push protocol.

III. Proposed Hybrid Protocol Design

The proposed protocol is designed for an ongoing bridge SHM project described in [6]. For this application, several types of sensor data are required to be collected via WSN using different sampling frequency:

- Generate one data every few minute: cable tension, steel/concrete stress, settlement, wind direction and speed, temperature, etc.
- Generate large amount of data every few hours: vibration.

Synchronized vibration nodes are required to generate over a thousand samples, where the whole data set can be used for high-degree spectral analysis of the structure. For better accuracy, the sampling amount should be as large as possible.

In order for the proposed protocol to work, the following conditions are needed:

- The WSN routing is set, that is, each node knows its parent and child nodes.
- The synchronization is properly conducted each time before the vibration sampling.
- Due to the limited memory size of the MCU, large vibration data are first stored in the on board flash chip.

The proposed design is mainly based on the TinyOS 2.x supported MAC protocol. The application requires all but the vibration data to be transmitted immediately for real-time analysis, therefore low power listening is not considered, the provided protocol becomes nothing but a simple CSMA/CA protocol. The collision avoidance cannot detect hidden node problem. To solve this problem, RTS/CTS are used to obtain the channel before data can be sent from transmitter to receiver. Nodes use RTS frame to compete the channel before send data. Receiver responds with a short CTS frame to confirm the request and allow the data to be sent. ACK frame is send by the receiver to confirm and finishes the transaction. Neighbour nodes who received either RTS or CTS go to sleep for the amount of time described in the RTS/CTS frame. The sleep time is long enough for the neighbours not to interfere the channel. To further increase the throughput of the network, protocol allows the neighbour nodes to wake up early and monitor the channel. If it receives ACK from the receiver, it means the transmission is over and this node can start its own transmission early.

For large data transmission, burst transmission protocol is used. After vibration data are sampled and stored in the flash, nodes compete for the channel using overhead frames. Consider a simple star network where several child nodes send data to the same parent. The following steps are taken.

1) Data Ready**:** The first handshake frame send from child nodes is "DataReady". It tells the parent node the sender's address along with number of data messages this child node is preparing to send.

2) Send requested data**:** The parent node receives all the child nodes' "DataReady" frame and stores the information in a list. Whenever it is ready, it will choose a node from the list and response with a "ReceiverRequest" frame. This frame tells the corresponding child node to send the data in bursts. Due to limited memory size of each node, one burst only allows a few data packets. The frame indicates the start and stop sequence of the data for this burst. Other child nodes received this frame go to sleep with the "Hold Time" indicated inside the frame.

3) Burst transmission**:** The sampled data are composed into one "TransferData" frame. For burst transmission, several "TransferData" are sent in sequence. Only the data sequences specified by "ReceiverRequest" frame are allowed to send. In each frame, "Hold Time" is also contained to allow unnecessary nodes to go to sleep.

4) Retransmission**:** After each burst transmission, the parent node checks for any missing sequence in the received buffer. If there is any missing, the sequence number is added to the "ReceiverRequest" frame for resend. The child node only needs to transmit that missing packet again. If all the sequences are received properly, "ReceiverRequest" frame is responded with an end signal.

5) Request for next sequence: For next burst transmission, another "ReceiverRequest" frame is sent with updated start and stop sequence numbers. After all the data are received, the parent node responses with "all data are received" signal using "ReceiverRequest" frame. The child node can now end its transmission and go to sleep.

6) Transmission of the next child node: The parent node selects another pending child node, and initiates another burst transmission by sending "ReceiverRequest" frame. The procedure is the same as shown above.

For complicated network where several child nodes need to send large amount of data to the sink through one or more relay nodes. The protocol becomes more complex. See Figure 1 as an example. Both node A and B has data ready to send. Their parent node is Relay1. Node B establishes the handshake with Relay1 and starts to transfer data as specified by Relay1 (step 1 through 4). Relay1 keeps the burst messages in the memory and establishes the connection to its parent node Relay2 and send out the data (step 1 through 4). Same procedures happens when Relay 2 send data to sink node. After one burst packet sequences are send to the sink node, Relay1 can ask node B to send the next burst packet sequences (step 5) and relay them to Relay2. After completely sending all burst data from node B, the transmission can move on to node A. If Relay1 has burst data as well, it will need to wait until node A and node B finished sending their burst data. The procedure stays the same.

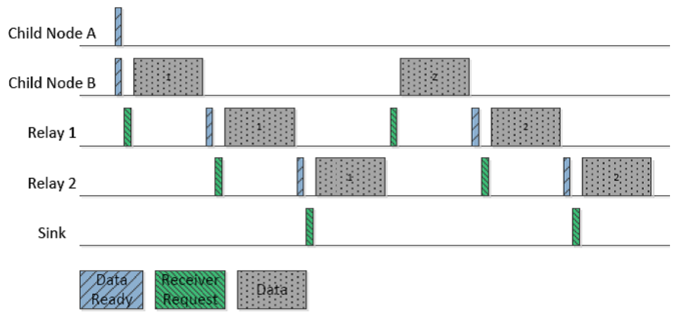

Figure 1. The typical time line of a burst transmission with 3 hop network

Since burst transmission consumes more power, it is set to have higher priority than CSMA/CA based low throughput transmission. Node samples data such as steel stress sensor compete the channel with vibration nodes will be lost and put into sleep. "Hold time" is embedded in all the burst transmission frames, any unwanted neighbour nodes can be put into sleep quickly. Therefore, interference is minimized. There is no end to end ACK check for the multi-hop transmission, only NACK is used after every burst (step 4) on every hop relay.

IV. Performance Analysis

The proposed hybrid protocol is implemented in TinyOS 2.x environment. The hardware we used is Micaz, which is supported by TinyOS 2.x. It has an Atmega128 MCU for processing and CC2420 transceiver for communication. It works on 2.4GHz frequency band with 16 channels. We tested the protocol on an indoor testbed. The testbed node also uses Micaz, but adds Ethernet ability for easy access (see Figure 2).

All the testbed nodes are connected to a Power Over Ethernet (POE) switch with an Ethernet cable that provides power as well as the ability to program and debug. The testbed nodes are installed on the celling of a lab room, with a layout of 4x4 grid. The distance between two nodes is about 2m. Ethernet cables are connected to a gateway; program can be uploaded through an internal webpage.

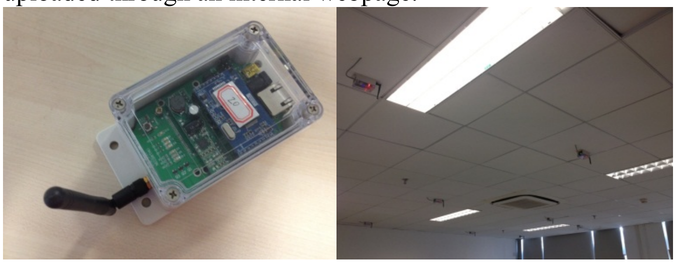

Figure 2. Left: Testbed node. Right: Testbed network installed on the celling of a lab room.

To test the burst transmission, some variables need to be set. The node generates 1024 data every time after synchronization. Since each data is very short, every data packet is compacted with 5 data. Due to the limited memory size, each burst transmission contains 10 data packets. Two end nodes are used to generate the data and the number of hops between end nodes and the sink varies from 2 to 6. Nodes use their maximum transmitting power to create an interference environment. For comparison purpose, a simple CSMA/CA protocol is used. It transmits the same amount of data with same packet length.

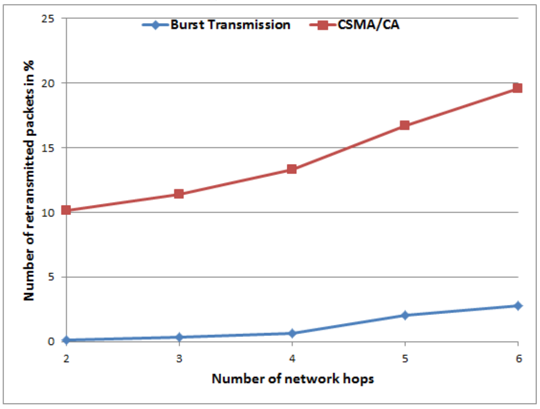

Figure 3. Packet retransmission rate for CSMA/CA and Burst Transmission

Figure 3 shows the packet retransmission rate in percentage. Since CSMA/CA has no hop by hop data lost check, the sink is responsible for checking the received data and asks for retransmission from the end node. It lost a lot of data and suffers from a heavy data retransmission rate, specifically when there are more relay nodes in the network. Burst transmission, on the other hand, has a very low data retransmission rate, even at a complicated multi-hop network environment. The test shows about 16% improvement at 6 hop network.

Due to the extra handshake frame used on the burst transmission, more time is consumed for data packets to be transmitted. Therefore, as number of relay nodes increases, the time duration for the transmission increases as well. Figure 4 shows that, although CSMA/CA suffers from a heavier data retransmission, it takes less time to finish the transmission when the network has more than 4 hops.

Energy consumption is also calculated (Figure 5). All the participating nodes' energy consumptions are added together. For maximum power consideration, the calculation uses the radio's receiving current provided by the CC2420 data sheet [16]: 18.8mA. The graph clearly shows that burst transmission protocol consumes much less energy. For CSMA/CA protocol, all the nodes stay awake even when they are not working. Large amount of energy is wasted due to the idle listening. Heavy packet retransmission rate also increases the energy consumption. On the contrast, the burst transmission put unnecessary nodes to sleep. No nodes in the network need to stay awake for the full operation time, but only wake up for their own transmission. Proper handshake overheads also lowered the data retransmission rate, thus limits the wasted energy. At 6 hop network test, the energy consumed is about 38% lower.

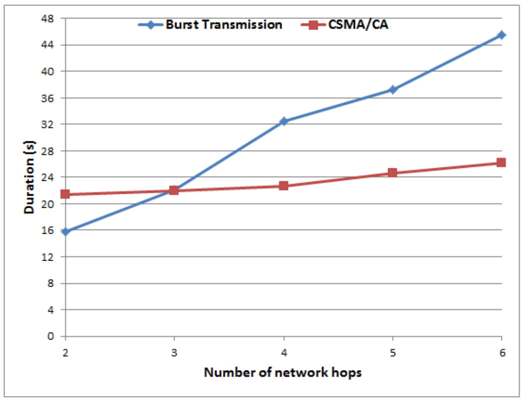

Figure 4. The transmission duration for CSMA/CA and Burst Transmission

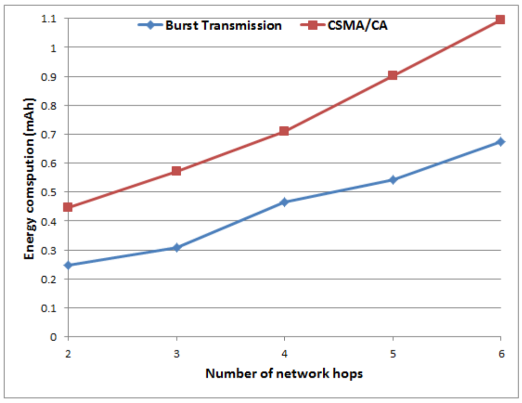

Figure 5. Energy consumption for CSMA/CA and Burst Transmission

The tests above set the size of burst packets to 10, to better accommodate the limited MCU memory size. Larger burst packet size is possible for which less control frames can be used, therefore reduce the operation time and power consumption. More tests and evaluations are ongoing. At the same time, the proposed hybrid protocol is also implemented on an ongoing bridge SHM project described in [6]. Its performance in outdoor environment is being tested. We will publish the results soon.

V. Conclusion

This hybrid protocol combines two MAC protocols together. The improved CSMA/CA protocol is suitable for slow data traffic and burst transmission protocol is a good solution for large data traffic. Channel competition is reduced to once when end nodes are ready to send all the packets. Hop by hop NACK instead of end to end NACK reduces the time and energy to transmit and relay extra packets back and forth, especially for multi-hop network. Overhead cost is very small, considering several long data packets share the same overhead frame sets. Hidden/expose node problem is also solved since the overhead frames put unnecessary nodes to sleep.

Testbed implementation shows very good results. By compare with the TinyOS supported CSMA/CA protocol, it greatly reduces the data retransmission rate. Although data transmission time is longer, nodes do not need to be awake for the entire operation, therefore save large amount of energy. More tests and evaluations are ongoing. At the same time, the proposed hybrid protocol is also implemented on an ongoing bridge SHM project. Its performance in outdoor environment is being tested. We will publish the results soon.

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] A. Bachir, M. Dohler, T. Watteyne and K. K. Leung, "MAC Essentials for Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials.,* vol 12, no. 2, second quarter 2010.

[2] A. Yin, B. Wang and Z. Liu, "Design and Implementation of the Structure Health Monitoring System for Bridge Based on Wireless Sensor Network," in *Proc. ISNN'09.* 2009, pp.915-922.

[3] S. Kim, S. Pakzad, D. Culler, D. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health Monitoring of Civil Infrastructures using Wireless Sensor Networks," in *Proc. IPSN'07,* 2007, pp254-263.

[4] J. P Lynch, Y. Wang, K. J. Loh, J. H. Yi, and C. B. Yun, "Performance Monitoring of the Geumdang Bridge using a Dense Network of High-resolution Wireless Sensors," in *Smart Material and Structures*, 15, 1561-1575, 2006.

[5] Z Zhao, J Sun, X Fei, W Liu, X Cheng, Z Wang, H Yang "Wireless Sensor Network Based Cable Tension Monitoring for Cable-stayed Bridges," in *Proc. ICACT'12,* Feb 2012, pp 527-532.

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

[7] *Part 11: Wireleess LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications,* IEEE Std. 802.11, 2007.

[8] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. INFOCOM'02,* June 2002, vol. 3. pp. 1567–1576.

[9] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. SenSys'04,* November 3-5 2004, pp. 95–107.

[10] M. Buettner, E. Yee, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *Proc. SenSys'06*, 2006, pp307-320.

[11] 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.

[12] C. Al. Boano, T. Voigt and N. Tsiftes, "Making Sensornet MAC Protocols Robust against Interference," in *Proc. EWSN'10*, 2010, pp. 272-288.

[13] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs),* IEEE Std.802.15, 2006.

[14] I. Rhee, A. Warrier, M. Aia, and J. Min, "ZMAC: a Hybrid MAC for Wireless Sensor Networks," in *Proc. SenSys'05*, 2-4 November 2005, pp511-524.

[15] D. Hasenfratz, A. Meier, M. Woehrle, M. Zimmerling and L. Thiele, "If you have time, save energy with pull," in *Proc. SenSys'10,* 2010, pp423-424.

[16] "CC2420 2.4GHz IEEE 802.15.4/ZigBee-ready RF Transceiver," Chipcon Products from Texas Instruments, Dallas, Texas, USA.