TELKOMNIKA, Vol.11, No.3, March 2013, pp. 1328 ~ 1337
ISSN: 2302-4046

B-MAC Design and Analysis for Embedded Sensor Networks

Liu Yumin*[1], Sun Yonghe[1], Xu Fengming[2], Wang Tao[2]
[1]School of Electrical Engineering & Information, Northeast Petroleum University,
[2]Daqing Oilfield Company, Development Street 199#, Gao xin District, 163318
*Corresponding author, e-mail: liuyumin330@163.com

Abstract

This paper presents medium access control (MAC) protocol designed for embedded sensor networks. I start with a protocol called MACAW, RTS-CTS-DS-DATA data packet exchange. Then I analyze the properties of some MAC protocols, including S-MAC, B-MAC and IEEE 802.15.4. According to the understanding of S-MAC, some advantages are summarized, such as energy saving, latency reduction, etc. however, there are still some disadvantages, including data packet lost, unfairness, synchronization. And then the similarities and differences between them are discussed. S-MAC is a novel technique to reduce energy consumption, but B-MAC is more efficient with long preamble. Using long preamble, B-MAC achieves collision avoidance and high channel utilization rate. B-MAC can minimize idle listening, but it needs bi-directional communication. Finally, the paper presents B-MAC design process and implementation result on MSP430.

Keywords: *B-MAC protocol, Embedded Sensor Networks, MSP430*

1. Introduction

Embedded sensor network is a network of embedded computers placed in the physical world that interacts with the environment[1]. Sensor networks have challenges in two key areas. First, energy consumption is a common problem in sensor network design. Most of sensors get power supply by batteries, and need to communicate with remote server; second, how sensors interact with its neighboring nodes within communication range effectively is another issue.

Media Access Control is an important component in embedded sensor networks communication process. Traditional strategy for dealing with packet collision avoidance is CSMA/CD- all nodes can tell between idle and busy link and they share the link. Some node listen as it transmits and can detect when collision occurs. Once the collision detection fails, they cannot detect collision and data packet will be lost. Thus, the channel bandwidth is wasted and energy utilization rate is very low. Based on the above consideration, the new MAC model is proposed. In this paper, I describe design of B-MAC with MSP430 microcontroller sensor node, and present the basic idea of communication process between sensor nodes. I discuss simulation process in details, including transmitter part, receiver part and other implementation issues.

2. Related Work

MAC is a wide field working in embedded sensor network and wireless communication field. In this part, I will introduce the basic idea of MAC, the properties of MAC protocol with energy saving-Sensor-MAC (S-MAC) and Berkeley-MAC (B-MAC) -a versatile low power MAC protocol for sensor network. While some comparison between B-MAC and IEEE 802.15.4 is done.

2.1. Multiple Access with Collision Avoidance (MACAW)

MAC is a technology to control when to send a packet and when to listen for a packet in wireless network. However, the idle waiting wastes amounts of energy. How to improve this technology is a new challenge in embedded sensor networks. Based on research on ad-hoc network, MACA protocol [2] is proposed, which is RTS-CTS-DATA scheme. To some degree, it

Received October 16, 2012; Revised January 1, 2013; Accepted January 23, 2013

solves the collision avoidance and coordinates the sensors communication. However, for some complicated case, it still cannot get better energy utilization. In [3], a novel technology- MACAW is proposed. It is a RTS-CTS-DS- DATA scheme. The technique details will be discussed as following, including the communication problems and how to solve them applying to new MACAW technique.

(1) Hidden Terminal Scenario

Figure 1. Hidden teminal problem

Hidden teminal problem shown as Figure 1.Node A wants to transmit message to node B, while C cannot hear node A, so C transmits data to B at the same time. In this case, collision happens at B. A and C are hidden from each other.

(2) Exposed Terminal Scenario

Figure 2. Exposed teminal problem

Exposed teminal problem shown as Figure 2.Node B sends to node A, while node C sends to other node, node D not node B. When C is ready to send, it detects B is transmitting something, so C defers its transmission. However, there is no reason to defer transmission because A is out of range of C. This problem is called exposed terminal problem.

（3）RequestToSend/ClearToSend (RTS/CTS)

The problems described above can be solved with an algorithm called Multiple Access with Collision Avoidance (MACA). This idea is for the sender and receiver to exchange control frame with each other before the sender actually transmits any data. This exchange informs all neighboring nodes that a transmission will start. Specifically, the sender transmits a Requesttosend (RTS) to the receiver; then the receiver replies with a Cleartosend (CTS). Any node that sees the CTS knows that it is close to the receiver and therefore cannot transmit for the period of time it takes to send data. Any node that sees the RTS but not the CTS is not close enough to the receiver to interfere with it. So it is free to transmit.

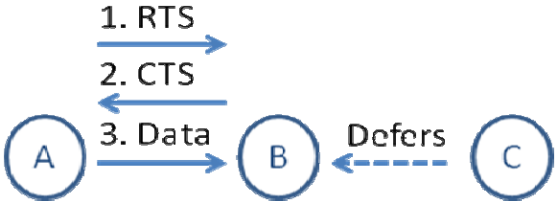

Figure3. Usage of RTS/CTS

When node A wishes to send to node B as shown in figure 3, it sends RTS to B. If B receives RTS, it immediately replies with CTS. Upon A receives CTS, it immediately sends data.

Any station overhearing RTS defers transmission until CTS has been finished. Any station overhearing CTS defers until data transmission is finished.

RTS/CTS avoid collision at the receiver, not sender. If station does not receive CTS, it will eventually be time out, which means a collision occurs.

(4) DS

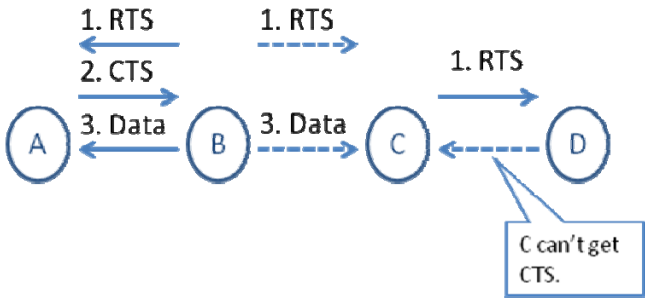

Figure 4. Collision Scenario for DS

DS is used to avoid the collision in exposed terminal scenario. For example, there are four nodes as follows:

Node B and C are in each other's range. So when B is transmitting data to A, A is the receiver. So theoretically at this time C can transmit to D, since A and D will not interfere with each other. However, in the traditional RTS - CTS - DATA mechanism C is not likely to hear the CTS from D, since B is transmitting data to A and interferes with C. Thus C can't get CTS from D, and may think that a collision happens. So C will back off and select a random time to restart sending RTS to D.

Generally speaking data packets are much larger than control packets. So the randomly selected retransmission time of C is unlikely to fall in the free time spans of B. In this way, while B is transmitting data to A, C probably will increase its back-off time greatly, and maybe will lose synchronization with D.

To solve this problem, B can transmit a DS packet before it transmits data, as the following figure 5 shows.

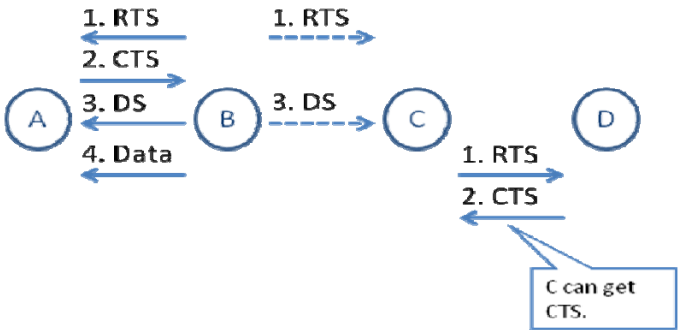

Figure 5. Usage of DS

In this case, when C hears the DS packet, it will know that B will be transmitting data, and it also will know the approximate time it will take B to transmit the data. And C can try to transmit RTS to D after B finishes its transmission of the current data packet, so that it can compete for media with B in the correct contention stage.

(5) RRTS

RRTS is proposed to solve the unfairness in a certain scenario of media access contention. The application scenario is shown in the following figure 6:

Figure 6. Collision Scenario for RRTS

In this scenario Nodes B and C are within each other's range. A tries to send data to B, and D tries to send data to C. And the data packets from any single link can saturate the network. If in the first place A and B win the contention, and begins the data transmission, then nodes C and D will likely be fully denied access to the media. This will happen like this: whenever D transmits a RTS to C, C will not respond with a CTS, because C will overhear the CTS from B and defer its transmission for B. Unless D happens to transmit the RTS in the relatively short time period between the completion of one data packet and the transmission of CTS from B, otherwise C will not reply a CTS to D.

To solve this problem the RRTS is adopted. Whenever C receives a RTS it can't respond due to deferral, it will contend during the next period and send a RRTS to D. Once D receives a RRTS it will immediately send a RTS to C, and hence C can reply with a CTS. And other nodes overhear a RRTS should defer for two slot times to hear whether a successful RTS-CTS has happened.

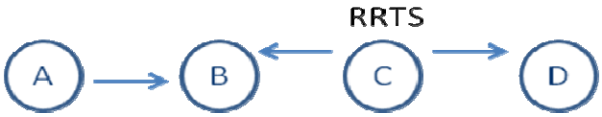

Figure 7. Usage of RRTS

In this way, the unfairness in this scenario can be solved. However, RRTS can't work in other scenarios, for example in the above figure 7.The A is transmitting to B, and C tries to transmit to D. Once C succeeds in transmitting to D, the transmission between A and B is unlikely to start, since at this time B can't hear RTS from A due to C's data transmission.

2.2. S-MAC (Sensor-MAC)

The core idea of S-MAC [4] is periodic listen and sleep in each node. Each node will keep a schedule of periodic listen and sleep, and it tries to get synchronized with its neighbors. During the listen period, the node can perform data transmission and reception if required. In sleep period the node can save energy by avoiding idle listening. In order to decrease the latency caused by unsychronous listen and sleep schedules between neighbor nodes, each node will try to get synchronized with its neighbors. In this way a pattern of coordinated sleeping is created among the neighborhood of each node. Active listening is introduced to this mechanism to reduce the latency in data transmission. And message passing is introduced to reduce the message-level transmission latency.

S-MAC is energy-efficient, [5] in that it can reduce the energy consumption in idle listening, overhearing, control packet overhead and collision. At the same time the sacrifice in latency is mitigated by the active listening and message passing. Though per-node fairness is also sacrificed, it's claimed to be not very important in WSN application scenarios.

Though S-MAC has many merits [6], it does have some inherent disadvantages introduced by the periodic listen and sleep. And some assumptions and claims may not seem very sound.

First of all, per-node latency is inevitably introduced by the periodic listen and sleep mechanism. If one node wants to transmit some data to its neighbor node which is still sleeping, the data transmission will fail, and it has to wait for the neighbor node to wake up in order to transmit data. In this way the per-node latency exists. Though active listening can reduce the latency, it can't be completely removed, since at least there maybe some chances that not all

neighbor nodes can overhear the RTS/CTS packets, or the timing for active listening does not match the actual data transmission.

Secondly, by using message passing the per-node fairness is sacrificed. Though in most WSN application scenario fairness is not so important, in some special scenarios it's very important. For example, if two nodes separately detect some different events and they both have to transmit messages (one (A) will transmit big messages and the other (B) transmit a small message for example an alarm message) via a common node (C). In this case, if unfortunately A gets control of media access in C first, it will occupy it for a long time. At the same time B only need a fraction of media access to transmit the alarm message. This is one of the cases in WSN applications where fairness is very important.

So, the extent of harm on per-node fairness introduced by S-MAC should be further investigated. And if it's possible, some boundary should be set on the deterioration of per-node fairness in S-MAC. Otherwise the sacrifice of per-node fairness may make S-MAC unsuitable for some special application scenarios.

Thirdly, in S-MAC the nodes still need to synchronize their sleep schedule with their local neighbors. SYNC packets have to be transmitted to reach this aim, and this adds up to the overall control packet overhead. And after sleep schedule synchronization, some nodes in the boundaries of the virtual clusters may have to adopt more than one schedule, and thus have to consume more energy than other nodes adopting only one schedule. This case may reduce the lifetime of some frequently-used nodes, and these nodes tend to be in the critical topology paths.

Fourthly, the fixed period of listening, together with the overhearing of data and control packets, still consumes energy. In active listen stage, one node may stay active for a fixed period without any data transmission or reception. This is a waste of energy. At the same time, though overhearing is reduced, it has not been completely removed. Overhearing of RTS/CTS packets and some data packets still will consume energy. And this means that something more can be done in order to further reduce energy consumption.

2.3. B-MAC vs. IEEE 802.15.4

As we know, B-MAC is a flexible benchmark MAC protocol [7] proposed to provide a small core of media access functionality, based on which variable mechanisms can be implemented to provide proper MAC functionalities for various Embedded Sensor Network applications. It's a test-bed protocol for research on WSN.

At the same time IEEE 802.15.4 is a set of protocols targeted at commercial applications of low rate WPAN [8]. It contains protocols for both PHY and MAC layers. To make the comparison between B-MAC and IEEE 802.15.4 reasonable and meaningful, here only the MAC functionalities in both will be compared [9].

Several similarities exist between B-MAC and IEEE 802.15.4:

(1) Beacon frames in 802.15.4 serves the similar aim as the Preamble in B-MAC. In 802.15.4 the super frame can be optionally enabled, which contains a beacon frame in the beginning. When network devices want to communicate with the coordinator, they will first get synchronized with the coordinate by listening to the beacon frame. And in B-MAC when a node wakes up from the sleep it will listen to the channel. If it gets the preamble, it will prepare to receive data frames.

(2) In 802.15.4, either beacon enabled or not, the network devices use CSMA/CD mechanism to contend for the channel resources. And in B-MAC, the Clear Channel Assessment (CCA) is also a kind of CSMA/CD mechanism.

(3) 802.15.4 can choose to use ACK in data transmission in order to provide reliable link. And B-MAC also uses link-layer ACK to enable reliable data transmission.

At the same time, great differences also exist between B-MAC and 802.15.4:

(1) Beacon frame in 802.15.4 is an optional feature. If it's disabled the network devices in the network will use CSMA/CA to contend channel resources. However, in B-MAC, the preamble is a must.

(2) In 802.15.4 beacon frame is very short, and in B-MAC the preamble should be longer than the active and sleep duration in order to make sure that the receiver node can get the preamble signal.

(3) In 802.15.4 when beacon is enabled, the full functionality device (FFD) can use GTS to reserve some time slots for their data transmission. And for the other time slots they will use CSSMA/CA to contend. However, in B-MAC there is no such a reservation mechanism.

(4) Frame security is provided in 802.15.4, while in B-MAC no security consideration is provided.

3. B-MAC Design Overview

In this paper, B-MAC is implemented on MSP430 eZ430-RF2500.The eZ430-RF2500 is a complete USB-based MSP430 wireless development tool providing all the hardware and software to evaluate the MSP430F2274 microcontroller, CC2500 2.4-GHz wireless transceiver, and highly integrated, ultra-low-power MSP430 MCU with 16-MHz performance [5].

The eZ430-RF2500 uses the IAR Embedded Workbench Integrated Development Environment (IDE) to write, download, and debug an application. The debugger is unobtrusive, allowing the user to run an application at full speed with both hardware breakpoints and single stepping available while consuming no extra hardware resources.

The SimpliciTI network protocol is designed for easy implementation with minimal microcontroller resource requirements.

B-MAC is a low power operation. Through this protocol, power can be saved. The key concept of this protocol is that it does not need synchronization. Before the transmitter sends messages, it should send a long preamble to wake up the receiver, and then actual data is sent. Thus, the receiver does not worry about that it will miss some messages. When transmitter prepares to send message, it has to make sure that the channel is clear, that is, no other noise message is being sent to the receiver in the network. Such a process is called Clear Channel Assessment (CCA).

When one transmitter is sending message to the receiver, another transmitter enters into the network. The current transmission is not interrupted, until it goes to sleep. At this time, another transmitter can send data to the receiver.

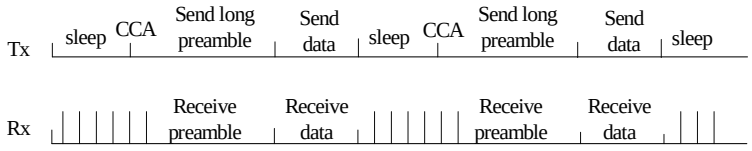

Figure 8. Timeline of Communication Process

From the Figure 8 above, the relationship between the transmitter and receiver can be indicated clearly.

4. Simulation Process

4.1. Transmitter Part

According to the B-MAC protocol definition, before Transmitter tries to send actual message, it needs to send long preamble. Thus, when receiver wakes up periodically, it can keep awake so that it does not miss receiving the actual message. And also, B-MAC uses Clear Channel Assessment (CCA) to check whether the transmission medium busy or idle. In this lab, this will be implemented in the simple way. The implementation process will be described in details as follows.

(1) Clear Channel Assessment (CCA)

For example, a node, T1, is mainly a transmitter. When T1 prepares to send the message to the destination, node R, it should firstly detect whether there is some other node sending the message (Noise Message). If so, node T1 should hold a moment and keep detecting the channel until it has been clear.

As for transmitter side, when the pushbutton on node T1 is pressed, it will send a message with value 0 and it stops sleeping and starts to send message. It firstly tries to receive something. In this case, if an indicator “Y” is received by Node T1, it means the channel is clear at present and communication between the Node T1 and Node R is safe and clear. However, when the Node T1 receives “N” over radio, even if it sends message, the collision will occur at Node R. in this situation, both Red and Green LEDs on T1 display on. And it will have to be held and check medium after some back off delay. Once the medium gets clear again, it will get resource and start its own transmission. The implementation flowchart is shown in Figure 9.

As for receiver side, once it receives some message, it should check what kind of message it received. If the message contains value 0, it means one transmitter sends a request for transmission. In this case, the receiver should further check whether the medium is busy or idle. If medium is busy, which indicates that some other transmitter is sending data to me. And then, the receiver sends “N” to indicate this situation. Otherwise, sends “Y” to indicate that the medium is clear, the corresponding transmitter can send data to me.

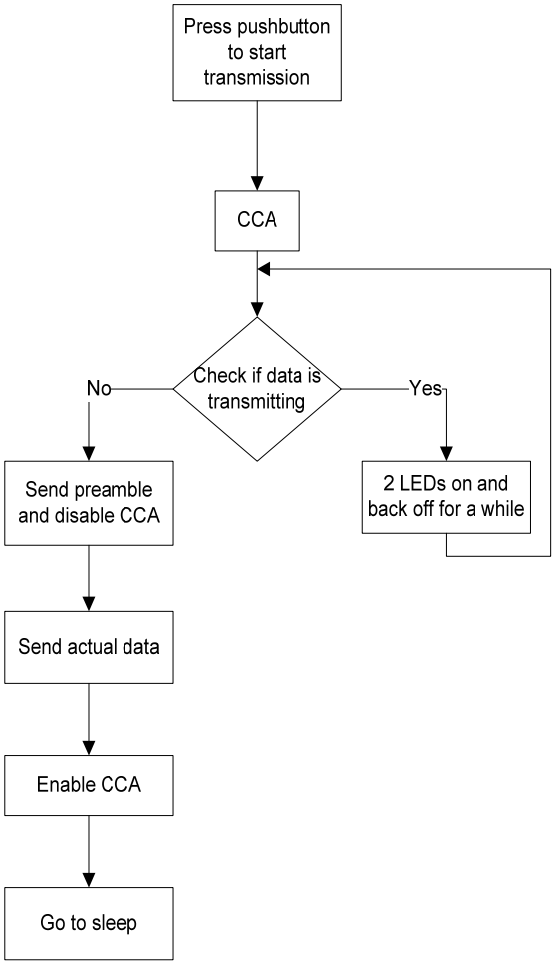

Figure 9. Flowchart of Transmission Process of B-MAC

(2) Sending Long Preamble

After CCA, long preamble is used to wake up the sleeping receiver. The time of preamble should be greater than receiver checking channel interval. Here, 5 times of check channel interval is set to make sure that receiver can wake up in time to receive actual message and minimize power consumption. For long preamble, constant value ox3F is set to avoid being

confused with other actual and meaning messages. When Transmitter is sending long preamble, the green LED is on until actual data is sent.

(3) Sending Actual Message

Once long preamble ends, actual message is going to be sent. The sent data can be any value except 0x3F (it is used for long preamble). When sending is done successfully, Red LED on transmitter is on. After sending the actual message, the transmitter enters sleeping state. The next transmission cycle does not start until the pushbutton is pressed again.

4.2. Receiving a Message

As for B-MAC protocol, receiver wakes up periodically, if no message is received, it will go to sleep again. However, once receiver receives long preamble, it means actual message is coming soon. It will not go to sleep but wait for accepting the messages.

For different message, receiver needs to takes different actions. When long preamble comes, the Green LED on receivers will be on. But if actual data is received, Red LED will be on.

After actual data is received, the receiver goes to sleep again and waits for later message. The implementation flowchart is shown in Figure 10.

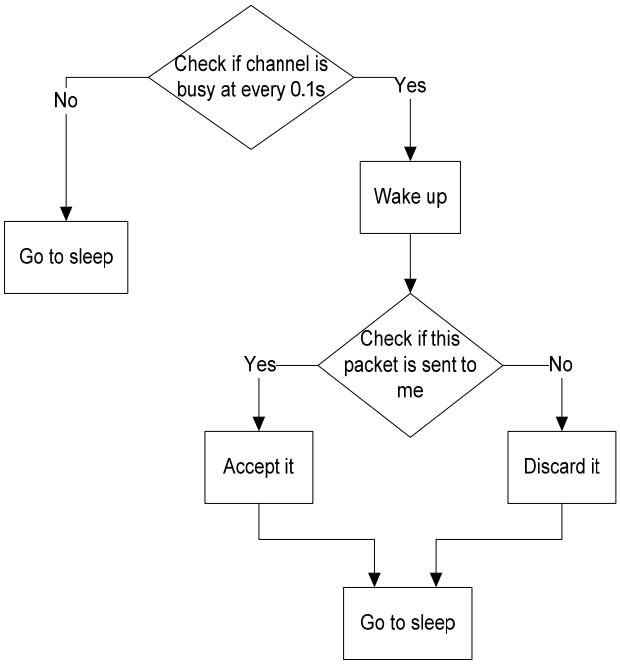

Figure 10. Flowchart of Receipt Process of B-MAC

4.3. Another Transmitter Enters into the Network

Once another transmitter, T2, enters into the network, when original transmitter, T1, is sleeping, the new transmitter will gain transmission right and new transmission cycle will start. At that time, when old transmitter wakes up, it cannot send any message out, so it has to hold a while until the channel clear. The details have been described in the implementation of CCA part.

In conclusion, if receiver is receiving some message from any transmitter, the other one will wait and also both of the LEDs display on to mark that channel is busy.

5. Experiment Result

When only one transmitter occurs in the network, the situation is very simple. Rx wakes up periodically and detects whether some message is coming. If so, receiver receives the long

preamble firstly, and then receives the actual message. The action of LEDs on both Transmitter and receiver is indicated in the following table 1.

Table 1.The Action of LEDs on Both Transmitter and Receiver

Action of T1	T1	Rx
Sleep	RED/GREEN off	Nothing
CCA	If channel is busy, RED/GREEN on	/
Send Long Preamble	GREEN on	GREEN on
Send Actual Data	RED on	RED on

When another transmitter T2, enters into the network, the graph looks a little more complex. The Timeline of Communication process of more than 1 transmitter is shown in Figure 11.Once some transmitter gains the transmission right, the other one will have to hold and keep checking the channel with exponentially back off.

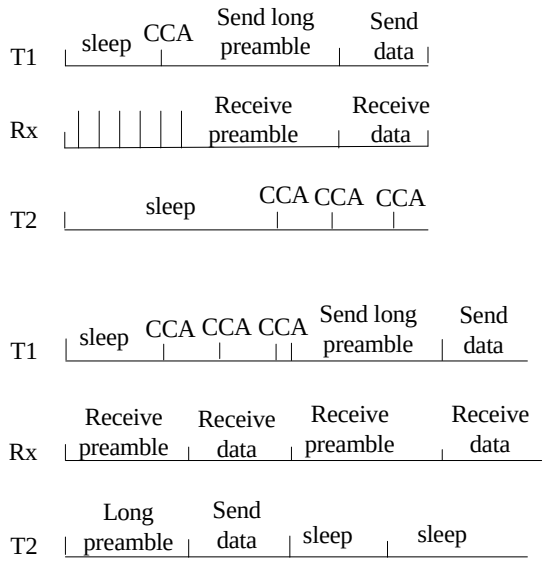

Figure 11. Timeline of Communication process of more than 1 transmitter

6. Conclusion

With the development of complex communication network, the requirement of stability and effectiveness for the sensor network is rissing. This paper presents the basic background about MAC protocol, especially MACAW, a new technology to control collision avoidance and effective communication derived from previous MACA protocol. S-MAC, an energy-saving protocol for ad-hoc networks is also presented. The periodical sleeping and waking mechanism is the merit of this protocol. B-MAC, as a new model outperforms other protocol, has low power utilization.

In this paper, B-MAC is implemented with MSP430 sensor node. We can see the simulation result and observe the performance of B-MAC. The experiment result follows the rule of B-MAC properties.

MAC protocol is an important component in communication process. Any improvement will benefit for the future embedded sensor network communication process.

References

[1] Wei Ye, John Heidemann, Deborah Estrin. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*. 2004.
[2] P Karn. *MACA—A New Channel Access Method for Packet Radio*. Proc. Ninth ARRL Computer Networking Conf. 1990.
[3] Hendra Setiawan, Yuhei Nagao, Masayuki Kurosaki, Hiroshi Ochi. *IEEE 802.11n Physical Layer Implementation On Field Programmable Gate Array*. 2012; 10(1).
[4] GJ Pottie and WJ Kaiser. Embedding the internet: Wireless integrated network sensors. *Commun. ACM*. 2000.
[5] A El-Hoiyi, JD Decotignie, and J Hernandez. *Low power MAC protocols for infrastructure wireless sensor networks*. In Proceedings of the Fifth European Wireless Conference. 2004.
[6] Xijun Yan, Xiangwei Meng, Yan Yan. A Wireless Sensor Network in Precision Agriculture. *TELKOMNIKA Indonesia Journal of Electrical Engineering*. 2012; 10(4): 788-797.
[7] JP Monks, V Bharghavan, and WW Hwu. A Power-Controlled Multiple Access Protocol for Wireless Packet Networks. *IEEE INFOCOM*. 2001.
[8] Joseph Polastre, Jason Hill, David Culler. *Versatile low power media access for wireless sensor networks*. Proceedings of the 2nd international conference on Embedded networked sensor systems. 2004.
[9] Shihong Duan, Yadong Wan, Peng Meng, Qin Wang. Hardware-in-the-loop and Parallel Simulation Architecture for Wireless Sensor Network. 2013; 11(1).
[10] MSP430F22x2, MSP430F22x4 Mixed Signal Microcontroller (Rev. D). 2010.