Low Power and Low Latency MAC Protocol: Dynamic Control of Radio Duty Cycle

Jeehoon Lee*†, Jongsoo Jeong†, Gyusang Shin†, Seon-Tae Kim†, and Pyeongsoo Mah*†

†Realtime Software Research Team, ETRI, Daejeon, Republic of Korea
*Department of Computer Science and Engineering, University of Science & Technology, Daejeon, Republic of Korea

Summary
In low-power and lossy networks (LLNs), power-constrained devices consume most of power to communicate with their neighbors. To save power, traditional MAC protocols usually turn off RF transceivers while there is no radio communication. Since RF transceivers consume most of their energy in idle listening, turning RF transceivers off can save significant amount of energy. However, delay of packet transmission is increased because the receiver sleeps most of time. It can lead a significant Quality of Service (QoS) decrease in protocols such as 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) and CoAP (Constrained Application Protocols). In this paper, we propose a low power and low latency MAC protocol to reduce delay while obtaining low power consumption. Our scheme makes an RF transceiver sleep most of time to save power and wake up periodically. When either receiving or sending packets occur, the wake-up period of RF transceiver shortens. In this paper, we compare our protocol with ContikiMAC, a low-power MAC protocol, by network simulation. The results of the comparison show that the latency of our protocol is lower than ContikiMAC. Our protocol also consumes similar amount of energy compared to ContikiMAC in the simulation.

Key words:
Low-power and Lossy Networks, Wake-up interval, Media Access Control, 6LoWPAN, CoAP

1. Introduction

Low-power and Lossy Networks (LLNs) are usually composed of a large number of wireless sensor nodes that have constrained power and resources. Accordingly, it is essential that wireless sensor nodes save their electric power. The nodes especially consume most of their power to communicate with other nodes by RF transceivers. Therefore, power of RF transceiver should be managed efficiently to achieve more energy saving.

There are many kinds of studies about power saving of RF transceiver. The energy efficient MAC protocols are divided into two categories, synchronous and asynchronous protocols. Synchronous protocols such as S-MAC [3] and T-MAC [4] negotiate wake-up time between sender and receiver nodes. They achieve low-power operation by reducing idle listening operation. However they have overhead for time synchronization. In contrast to synchronous MAC protocols, asynchronous MAC protocols make nodes keep only their own duty-cycle, so that they do not require any synchronization related overheads. However, the network latency is increased because the packet cannot be delivered to the destination node immediately until the node turns on RF transceiver and is ready to receive packets.

In conventional wireless sensor networks that collect sensing data, latency is not an important factor relatively because most of data can be contained in a single packet, and most of traffic patterns are only destined for a single point, e.g., sink. However, communication protocols of modern Internet of Things are different from them. Some IPv6 packets may be fragmented by 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) due to their bigger size than link maximum transmission unit (MTU). Quality of Service (QoS) of web-based packets such as CoAP (Constrained Application Protocol) [6] that are commonly exchanged in Internet of Things (IoT) environments also may be exacerbated. Since request packets require immediate response packets, long latency may cause time-out retransmissions. As a result, overall QoS can be significantly decreased.

For this reason, we propose a Low Power and Low Latency MAC (LPLL MAC) protocol to mitigate network latency with low energy consumption. LPLL MAC adjusts wake-up interval of RF transceiver according to network traffic to receive packets faster. When packet exchanges occur, the wake-up interval becomes short for defined time. After the time expires, the wake-up interval returns to the previous interval. When nodes receive or send a packet successfully, their wake-up intervals are shortened to receive next packets more quickly. LPLL MAC mitigates latencies especially in modern IoT protocols such as 6LoWPAN and CoAP. It consumes similar amount of energy compared to other low power MAC protocols.

This paper is organized as follows. Section 2 presents researches related to low power MAC protocols. In section 3, we introduce a mechanism about LPLL MAC protocol and show implementation issues such as timing. In section 4, we evaluate our scheme. The results show that our scheme reduces network latency and consumes similar amount of energy compared to ContikiMAC [1], a radio

Manuscript received December 5, 2012
Manuscript revised December 20, 2012

duty cycling MAC protocol. We conclude the paper in Section 5.

2. Related Works

Since energy efficiency is necessary for long life of LLN, a variety of low power MAC protocols have been studied recently.

ContikiMAC is an asynchronous and energy efficient radio duty cycling MAC protocol as shown in Figure 1. RF transceiver is kept off most of time and wake up about 1% of duty cycle. In this protocol, a node transmits a same packet until a receiver wakes up and receives the packet. This protocol also uses phase lock and fast sleep techniques which make RF transceiver turn off for longer period of time. However, the MAC protocol shows high network latency due to the short activation period of RF transceiver.

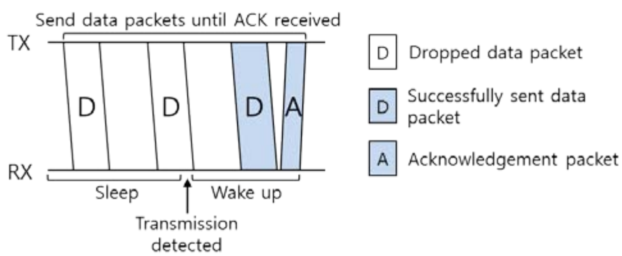


Fig. 1 Principle of duty cycling protocol. Sender TX transmits a same data packet repeatedly until it receives an acknowledgement packet. Receiver RX wakes up and detects radio signal. Then RX maintains wake-up state to receive packets and send acknowledgement packets.

X-MAC [2] is also an asynchronous duty cycle MAC protocol. It applies short preamble that makes nodes that are not targets go back to sleep fast and strobed preamble to allow the target node to interrupt sending preamble as soon as it wakes up. It saves energy consumption and reduces latency by the short preamble sending, but delay from duty cycle also occurs in case of sending multiple frames continuously.

In addition, low-power MAC protocols are required to be adopted to IoT network protocols. There are some researches about the interoperability between ContikiMAC and TinyOS LPL [7]. It shows that configuring parameters such as transmission and wake-up intervals has a decisive effect on performance.

There is also a study about applying ContikiMAC with CoAP, an emerging application layer protocol for IoT [8]. ContikiMAC shows energy efficiency and it is well performed with CoAP, but the latency of ContikiMAC is significantly increased.

For Supporting QoS over multihop wireless mesh networks, multichannel time-division multiple-access media access control (McTMAC) protocol [11] is proposed. McTMAC protocol reduces the end-to-end delay by longest flow first (LFF) channel and time-slot allocation algorithms. However, since it is a TDMA-based MAC protocol, it may not be practical to both implement and operate over long time due to the time synchronization overhead.

3. Design and Implementation

The principle of our protocol is based on ContikiMAC, an asynchronous radio duty cycling protocol. In addition to this, we applied a simple duty cycle control method in our protocol to reduce latency.

3.1 Principle of Radio Duty cycling MAC Protocol

The main idea of radio duty cycling protocol is that RF transceivers are turned off most of time and turned on periodically to check the channel by Clear Channel Assessment (CCA) whether there are radio activities. If activities are detected, RF transceivers maintain wake-up state and listen to receive a frame. Otherwise, RF transceivers return to sleep state again. After a receiver receives a data frame correctly, it sends an acknowledgement frame to the sender if required.

On the other hand, the sender just transmits a same packet repeatedly until it receives an acknowledgement frame. This scenario is shown in Figure 1. If the node cannot receive an acknowledgement frame over the sleep duration, it stops waiting for acknowledgement and is forced to turn off its RF transceiver. In case of broadcast or transmission without requirement about acknowledgement frame, it sends a packet repeatedly during the full wake-up interval.

3.2 Detail of LPLL MAC

In IoT environments, both request-response and continuous packet transmission patterns are common. Since these traffic patterns consist of two or more packet exchanges, latencies can be increased in low-power duty cycling MAC protocols. And the latencies may be proportional to the wake-up interval of RF transceivers. So the main idea of this paper is to change the radio duty cycle dynamically to mitigate latencies.

The scenario of changing wake-up interval is shown in Figure 2. Receiver RX sleeps during long wake-up interval (t_{lw}) at first. When RX receives a data packet after it wakes up and sends an acknowledgement packet, the wake-up interval is changed to short wake-up interval (t_{sw}). When the sender sends a packet and receives an acknowledgement correctly, LPLL MAC protocol shortens the wake-up interval of the sender. As a result, response or successive frames are transferred more quickly as the ratio of channel check is increased. In case of broadcasting, the wake-up interval is not changed because the sender does not need to wait for response messages.

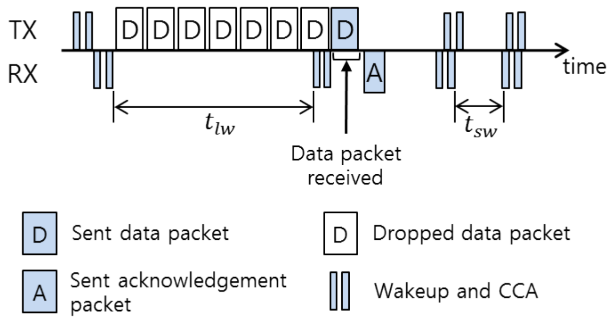


Fig. 2 A scenario of changing wake-up interval

A receiver also decreases its wake-up interval when the valid packet is received. The reduced interval is same as the short interval of a sender. It is possible that destination nodes also wake up more frequently for fast reception of successive packets.

The increased duty cycle consumes more energy than the traditional way while mitigating latency. In order to minimize the overhead of energy consumption and reduce latency at the same time, the short wake-up interval should be maintained only when radio traffic is expected to occur. Since we focus on the patterns of successive packet exchanges, the short interval is maintained while following expected packet exchanges occur.

In order to control the radio duty cycle, we implemented an individual wake-up timer. When the timer expires, the node wakes up and performs CCA. To change the wake-up interval, we configure two states, long wake-up and short wake-up state. When the node is set to a short wake-up state, short wake-up duration timer is initialized and run. When the timer expires, the node returns to a long wake-up state and wake-up timer is also changed to the long interval.

It consequently reduces latency about the traffic that already occurred on the same path before. However, the latency about the first packet transmission is not reduced because the wake-up interval is still long at the first time.

4. Evaluation

For our experiments, we implement LPLL MAC protocol in NanoQplus [5][9] which is a multi-threaded operating system for IPv6-based wireless sensor networks.

In order to evaluate our scheme, we simulate LPLL MAC protocol by COOJA [10], the Contiki network simulator, as shown in Figure 3. COOJA is able to execute programs by running compiled program code in an instruction-level emulator and simulate nodes running another operating system. It allows NanoQplus programs to be executed in COOJA simulator. In simulations, we deploy virtual nodes of Tmote Sky. The RF transceiver used by the Tmote Sky is the Chipcon CC2420, which is an IEEE802.15.4 compliant device. The mote also uses an 8MHz TI MSP430.

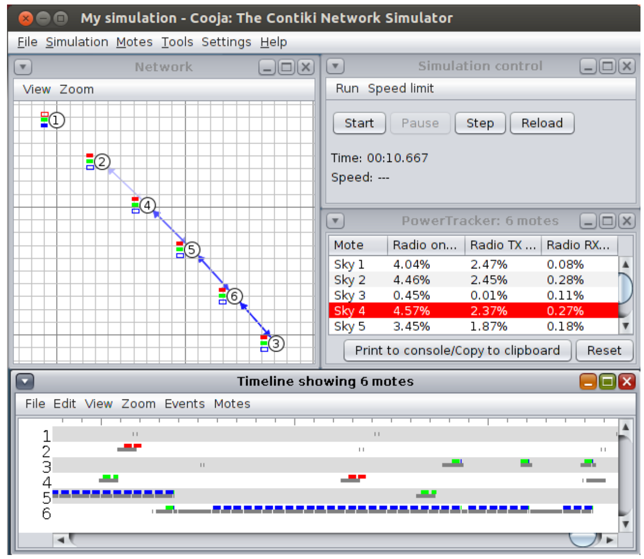


Fig. 3 User interface of COOJA network simulator

Through simulations, we evaluate LPLL MAC in terms of energy consumption and network latency. LPLL MAC is also compared with ContikiMAC without phase lock and fast sleep as a fixed duty cycle protocol. We set the duty cycle of ContikiMAC to 8 Hz. For LPLL MAC protocol, we set its low duty cycle to 8 Hz, and high duty cycle to 32Hz. The long wake-up state is maintained for one second after the node receives a correct data or an acknowledgement frame.

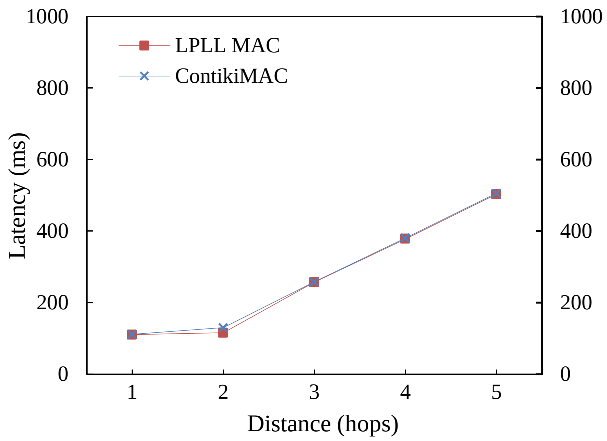
In addition to simulations, we port LPLL MAC and ContikiMAC to the real mote, UBee430, to find out real energy consumption. The UBee430 is developed at the HUINS and the mote uses TI MSP430 and Chipcon CC2420.

4.1 Latency

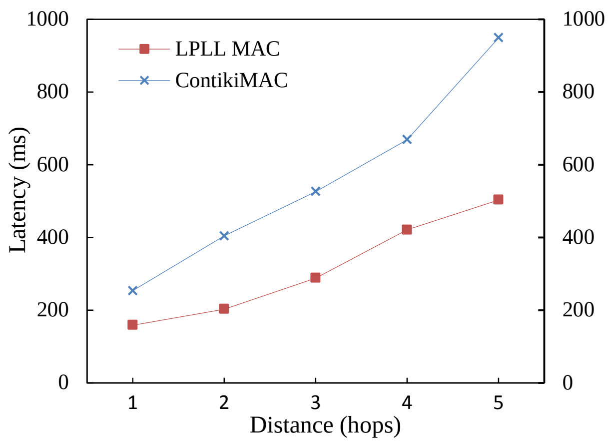
Network latencies are measured in terms of one-way latency and round-trip latency. We measure both kinds of latencies of LPLL MAC and ContikiMAC protocol. The latency is averaged by transmitting a hundred of packets, and compared according to the distance and the number of frames transmitted in succession.

We measure round-trip and one-way latencies according to the distance of destination. The results are shown in Figure 4. In the simulation, the source node transmits a 127-byte packet to the destination and receives a reply packet.

This is performed 100 times for every different distance and averages are calculated. In Figure 4(a), the one-way latency times between LPLL MAC and ContikiMAC protocols are almost same because the nodes of two protocols are all set to 8Hz duty cycle while there is no traffic. However, the latency times about receiving a reply packet from the destination are significantly different between them as shown in figure 4(b). Since LPLL MAC protocol change nodes' duty cycle to 32Hz when the nodes send or receive a packet, a reply packet is received more quickly than the ContikiMAC. The result shows that LPLL MAC protocol can decreases round-trip latency.



(a) One-way latency



(b) Round-trip latency

Fig. 4 Latency about transmission of one packet with different distance

Another simulation is to measure latencies about transmitting multiple frames in succession. It assumes that a transmitted packet is fragmented into several frames. Figure 5 shows the one-way latency of fragmented packet transmission. In case of LPLL MAC protocol, the one-way latency is increased a little even though the number of transmitted frames is increased. In contrast, the latency of ContikiMAC protocol is increased drastically. Through this simulation, LPLL MAC protocol seems to be suitable for the environment that large packets are fragmented.

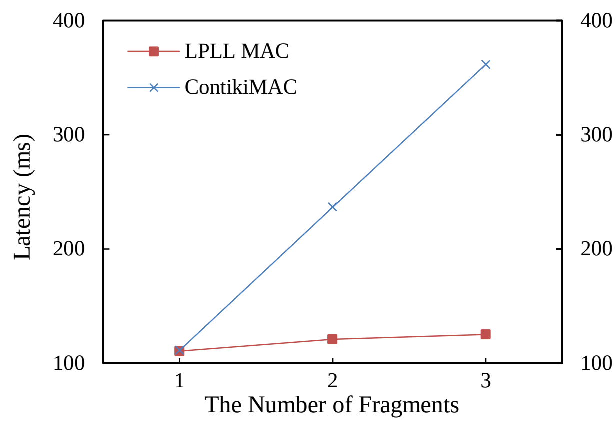


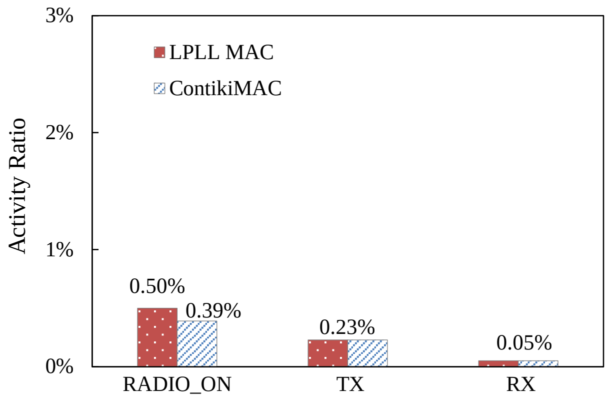
Fig. 5 One-way latency about transmission of fragmented packets in a single hop topology

4.2 Energy Consumption

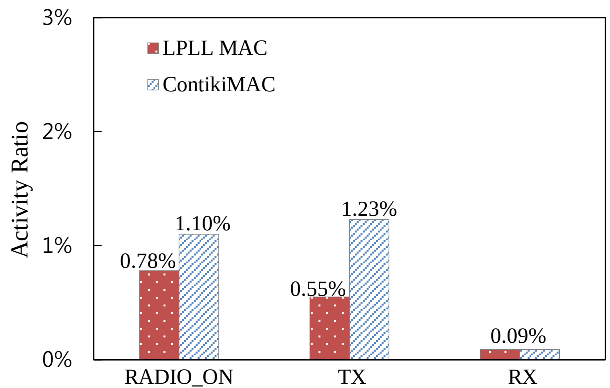
To compare the energy consumption of LPLL MAC and ContikiMAC protocol, we measure the activity ratio of RF transceiver. The activity ratio of both LPLL MAC and ContikiMAC protocol are same when there is no radio traffic. However, when packets are forwarded, the activity ratio of LPLL MAC protocol is higher than that of ContikiMAC because LPLL MAC makes RF transceiver wake up often after transmitting or receiving a packet.

Figure 6(a) shows the results of the worst case of LPLL MAC protocol. In the worst case, one packet is delivered while the short wake-up interval is maintained with no reply packet. A source sends a packet to the destination in a single hop topology and does nothing during the short wake-up interval. The ratio of transmission and reception of both protocols are same, but LPLL MAC protocol activates RF transceivers only 0.11% longer than ContikiMAC protocol due to the shortened wake-up interval after packet exchanges. Since this overhead is negligible, LPLL MAC protocol shows similar energy consumption compared to ContikiMAC protocol.

In case of multiple frame transmission, the LPLL MAC protocol is better than ContikiMAC as shown in Figure 6(b). When two frames are transmitted in succession, LPLL MAC can catch the second frame earlier than ContikiMAC. Eventually, this causes decrease of transmission ratio and average ratio of RF transceiver-on, although LPLL MAC protocol makes RF transceiver wake up more frequently than ContikiMAC protocol.



(a) Transmitted one frame



(b) Transmitted two frames in succession

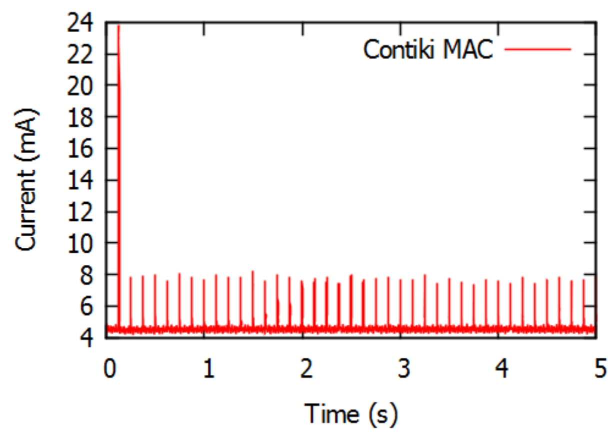
Fig. 6 Duty cycle of radio transceiver in a single hop topology

Through the results of simulations, we found that LPLL MAC protocol reduces the latency significantly in case of packet fragmentation. This means LPLL MAC protocol can be combined well with 6LoWPAN that fragments an IPv6 packet into suitable size of frames. In addition, LPLL MAC can also be applied well with CoAP, an application layer protocol, because the low latency of LPLL MAC protocol can increase the QoS using CoAP.

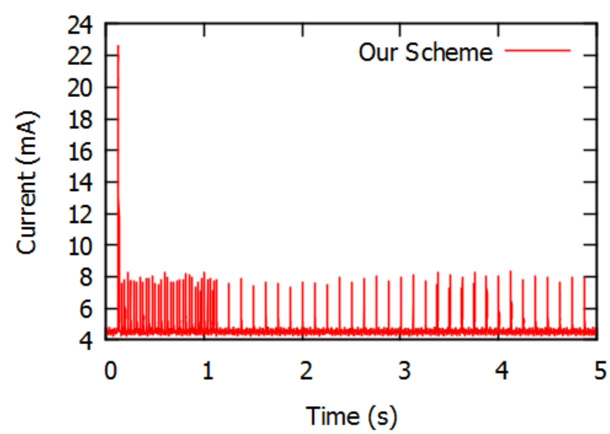
In case of energy consumption when transmitting one frame, LPLL MAC protocol consumes more energy than ContikiMAC protocol. However, in case of transmitting two successive frames, LPLL MAC protocol saves more energy than Contiki MAC protocol. Therefore, LPLL MAC protocol saves similar energy consumption compared to ContikiMAC protocol.

In order to show the actual power consumed, we attach the NI USB-6216, a data acquisition (DAQ) device, to the UBee430 mote, and measure the current draw with the DAQ by measuring the voltage over 20Ω resistor connected with a power supply.

Figure 7 shows the current draw of a packet reception in ContikiMAC, and LPLLMAC protocol during 5 seconds. In figure 7(a) ContikiMAC receives a packet and wakes up every 125ms. In figure 7(b), On the other hand, LPLL MAC wakes up every 31.25ms for one second after receiving a packet and then its wake-up interval is changed to 125ms like ContikiMAC. Table 1 shows the total energy consumption of Figure 7. The LPLL MAC consumes 0.16mJ of energy more than ContikiMAC which means that LPLL MAC protocol consumes similar energy compared to ContikiMAC protocol.



(a) ContikiMACe



(b) LPLL MAC

Fig. 7 Current Duty cycle of radio transceiver in a single hop topology

Table 1: Comparison of the energy consumption of the packet reception

MAC Protocol	***Energy (mJ)***
ContikiMAC	76.40
LPLL MAC	76.56

4.3 Results

Through the results of simulations, we found that LPLL MAC protocol reduces the latency significantly in case of packet fragmentation. This means LPLL MAC protocol can be combined well with 6LoWPAN that fragments an IPv6 packet into suitable size of frames. In addition, LPLL MAC can also be applied well with CoAP, an application layer protocol, because the low latency of LPLL MAC protocol can increase the QoS using CoAP.

In case of energy consumption when transmitting one frame, LPLL MAC protocol consumes more energy than ContikiMAC protocol. However, the difference of amount of energy consumption is negligible and LPLL MAC protocol saves more energy than Contiki MAC protocol in case of transmitting two successive frames. Therefore, LPLL MAC protocol saves similar energy consumption compared to ContikiMAC protocol.

5. Conclusions

ContikiMAC protocol is an energy efficient protocol, but it does not consider network latency. In contrast, LPLL MAC protocol is an energy efficient protocol and it also mitigates latency by controlling radio duty cycle. These advantages are more important to modern IoT protocols such as 6LoWPAN and CoAP. In addition, LPLL MAC protocol has less overhead to operate because it does not require negotiation between senders and receivers.

In the future, we will deploy a large number of nodes with LPLL MAC protocol to construct large scale IPv6 networks and see how it works in the real world.

Acknowledgments

This work was supported by the Industrial strategic technology development program, 10035570, Development of self-powered smart sensor node platform for smart & green buildings, funded by the Ministry of Knowledge Economy (MKE), Korea.

References

[1] A. Dunkels, “The ContikiMAC radio duty cycling protocol,” Technical report T2011:13, Swedish Institute of Computer Science, Decembers, 2011.

[2] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han, “X-MAC: A short preamble MAC Protocol for duty-cycled wireless sensor networks,” In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pages 307–320, 2006.

[3] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” In Proc. IEEE INFOCOM, New York, NY, pp. 1567-1576, June, 2002.

[4] T. van Dam, and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Nov, 2003.

[5] S. C. Kim, H. Y. Ki, J. K. Song, M. S. Yu, and P. S. Mah, “NanoQplus: A multi-threaded operating system with memory protection mechanism for wireless sensor networks,” In 1st China-Korea WSN Workshop (CKWSN), Chongqing. China, 2008.

[6] Z. Shelby, K. Hartke C. Bormann, and B. Frank, “Constrained Application Protocol (CoAP),” draft-ietf-core-coap-11, July, 2012.

[7] J. G. Ko, N. Tsiftes, A. Dunkels, and A. Terzis, “Pragmatic low-power interoperability: ContikiMAC vs TinyOS LPL,” In Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on, pages 94-96, June, 2012.

[8] M. Kovatsch, S. Duquennoy, and A. Dunkels, “A low-power CoAP for Contiki,” In Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, pages 855-860, Oct, 2011.

[9] J. S. Jeong, J. S. Kim, and P. S. Mah, “Design and implementation of low power wireless IPv6 routing for NanoQplus,” In Advanced Communication Technology (ICACT), 2011 13th International Conference on, pages 966-971, Feb, 2011.

[10] A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with COOJA,” In Local Computer Networks, Proceedings 2006 31st IEEE Conference on, pages 641-648, Nov, 2006.

[11] 1] Tran Minh Trung, and Jeonghoon Mo, “A Multichannel TDMA MAC Protocol to Reduce End-to-End Delay in Wireless Mesh Networks,” ETRI Journal, vol.32, no.5, pp.819-822, Oct, 2010.