Enhancement of Asynchronous MAC Protocol in Wireless Sensor Networks for Smart Monitoring Applications

Vo Que Son and Tran Truong Son

Abstract—Up to now, many research works have been focusing on designs in MAC layer of Wireless Sensor Networks (WSNs) to target high throughput, low latency and solving the problem of idle listening. In this paper, an enhancement of low-power asynchronous protocol for MAC layer, namely E-MAC, is proposed to target a long lifetime of sensor nodes working 6LoWPAN/IPv6 networks. Moreover, the E-MAC protocol is implemented on TI low-power platforms to verify the efficiency of the proposed design. Several parameters such as energy consumption, latency and packet delivery ratio are simulated and measured in a real monitoring test-bed to show the better performance in comparison with other popular MAC protocols.

Index Terms—Wireless Sensor Networks, MAC, monitoring applications, duty-cycling, low power.

1. Introduction

In wireless networks, a common radio channel is shared by many nodes via free space. However, if more than one node try to access this medium, collision will occur. From the energy aspect, the effect of collision in WSNs on the overall performance will decrease the channel capacity due to the corrupted packets and waste of energy because neighboring nodes may be involved in idle listening.

Energy efficiency is a key target in designing communication protocols for WSNs (e.g. routing and MAC layer protocols). One of the primary mechanisms in WSNs to achieve low energy consumption is duty-cycling. In this approach, sensor nodes periodically switch between awake and sleep states, in which the duty cycle period of a sensor node is equivalent to its awake time plus sleep time. Because of modern current RF hardware platforms using CMOS, the current consumption in sleep state is much less than that in the awake state (e.g. CC2420 platform [1]) if nodes which sleep most of the time can consume less energy. With a given duty cycle, challenges in protocol design for WSNs is to achieve high successful packet delivery rate, high throughput and low delay.

One of the problems in MAC protocols is idle listening time, in which a node has to wake up to check the incoming packets in the radio channel which may not be destined to itself. Hence, this idle listening wastes nodes energy. Many standard MAC protocols developed for WSNs [2-7] are motivated to reduce the idle listening time. These protocols can be categorized into three approaches:

- Synchronous protocols, belonging to TDMA scheme, require the schedule negotiation among nodes by exchanging frames. This schedule is defined when nodes are awake and sleep. Famous candidates for this kind of protocols are S-MAC [2], T-MAC [3], TRAMA [8], NAMA [9], TSMP [24]. However, TDMA protocols suffer from deficiencies when applied to WSNs in aspects such as network scalability, mobility, broadcasting, etc.

- Asynchronous protocols like B-MAC [4], X-MAC [5], ContikiMAC [7], RI-MAC [12], A-MAC [25] rely on low-power listening (also called preamble sampling) to link the sender to a receiver which is duty cycling. This mechanism shifts the synchronization to the sender. When a sender has data, it transmits a preamble, which is at least as long as the receivers sleep period. After waking up, the receiver can detect the preamble and stay awake to receive the data. This allows sensor nodes to communicate with each other without explicit exchanging synchronization information among nodes. The idle listening problem reduces because receiving node only wakes up for a short time to sample the medium. However, the length of the preamble is a challenge in designing asynchronous protocols, especially in non-packetizing radios [5].

- Hybrid protocols such as WiSeMAC [6], SCP-MAC [10], and Z-MAC [11] try to overcome the disadvantages of TDMA MAC protocols by combining TDMA with certain types of asynchronous support.
From all literatures mentioned above, our goal is to design a MAC protocol that can combine advantages of both synchronous and asynchronous approaches. Therefore, the proposed E-MAC has several key points as follows:

- Support low duty cycle and power consumption on nodes.
- Allow self-synchronization in nodes.
- Target high throughput and Packet Delivery Ratio (PDR).
- Utilize packetizing radio at PHY [5] and target good performance in networks with heavy load.

This paper is structured in 5 sections: section I is the introduction to the scope of this paper. The proposed design of E-MAC is deeply discussed in section II. Implementation of E-MAC is mentioned in Section III while Section IV describes the simulation and experiment results to validate the proposed model. Finally, conclusions are given at the end of this paper.

2. Design of Enhanced-Mac (E-Mac)

2.1. Problems in other Asynchronous MAC protocols

Firstly in BMAC [4], the receiver usually has to wait until finishing the long preamble to start the packet reception although the receiver wakes up at the beginning of the preamble sequence. This causes the waste of energy consumption for the sender because it has to finish the preamble transmission. Secondly, when the transmitter sends the preamble, other neighboring nodes are unexpectedly waken up and they also have to wait until the end of preamble to know whether the current packet is destined to themselves or not. If the current transmitting packet is not sent to themselves, they can enter the sleep mode at this point of time. This operation increases the energy consumption due to the idle listening problem, especially when the network traffic is heavy, all nodes seem to be in the active mode for listening. Finally, both sender and receiver have to wait until the end of preamble to start the transmission and reception phase, leading to a longer packet delay.

X-MAC [5] improves the problem of long preamble sequence in B-MAC by using a short preamble (wake-up sequence). However, this MAC protocol also has disadvantages such as the constraint of receiving the latest wake-up packet at the receiver and long listening time for the receiver (e.g. CC2420 running X-MAC has the channel checking period of 6.25ms). Besides, X-MAC uses a hand-shaking procedure between sender and receiver. If a receiver can correctly detect a preamble and returns an ACK to the sender, the sender cannot receive this ACK due to an error. Therefore, the hand-shaking procedure is considered a failure and it has to transmit a preamble again. Moreover, the preamble sequence used in X-MAC is constructed as a frame. It does not use the application data for synchronization, which results in the low throughput of X-MAC.

ContikiMAC [7] tries to solve all the disadvantage of [4, 5] by using a very short CCAs for detecting the radio activity. This improvement achieves the low duty cycle operation of sensor nodes. However, the very short CCAs also cause the missing packet transmission and generate an unstable channel indication.

2.2. Operation of E-MAC

E-MAC falls into the asynchronous MAC protocol, in which the sender will start the packet transmission and receiving node will use the CCA technique to detect the transmission. In E-MAC, nodes sleep most of the operation time and periodically wake up to check for radio activities. If a packet transmission is detected, the receiver stays awake to receive the next packet and sends a link layer ACK. To send a packet, the sender repeatedly sends the same packet until a link layer ACK from a corresponding receiver is returned. The visual strategy of E-MAC is illustrated in Figure 1 for comparison with other asynchronous MAC protocols.

![Fig. 1: Visual strategy of popular MAC protocols and E-MAC.](image)

With the broadcast transmission illustrated in Figure 2, the receiver does not need to return an ACK to the sender; hence, the sender repeats the packet transmission in a long interval so that all the neighboring nodes can receive this packet.
Different from other popular MAC protocols [5, 7], E-MAC improves the wake-up mechanism and simplifies channel listening technique to reduce the timing constraints and the processing load for MCU. Besides, it enhances the faster phase-lock of neighboring nodes to decrease the packet latency and energy wasted by idle listening time. These are discussed in the following sections.

2.3. Wake-up mechanism

E-MAC senses the channel activity based on CCA (Clear Channel Assessment). CCA uses the Received Signal Strength Indicator (RSSI) at the receiver to determine the idle or busy state of the radio channel. Theoretically, the CCA timing must be designed as short as possible so that nodes can save energy when there are no transmission on radio channel.

Unlike ContikiMAC [7] which uses two short CCAs for detecting the channel states (idle or busy), E-MAC does not use 2 separate CCAs to avoid the frequent change between radio transceivers states. These two short CCAs can reduce the consumed energy but also increase the possibility of missing packet transmission. Moreover, frequent switching between modes, especially switching from a sleep mode to an active mode, leads to more energy consumption than leaving the radio transceiver unit in idle mode because of the start-up power [23].

Therefore, E-MAC uses a longer reliable CCA with multiple RSSI checks (shown in Figure 3 and Figure 4) to estimate the channel states, which can slightly increase the power consumption but can detect the packet transmission faster. From a general point of view, this mechanism can help sender and receivers reduce energy because they do not tend to miss any packet transmission.

2.4. Timing constraint

Channel checking mechanism of E-MAC is illustrated in Figure 4 with several timing definition parameters as follows:

- $t_i$: the interval between each packet transmission, which can be chosen by implementation.
- $t_r$: the time required for a stable RSSI, depending on the hardware specification.
- $N$: the number of RSSI check in periodic wake up which can be configured by the implementation.
- $N^*t_r$: the time of each periodic wake up.
- $t_a$: the time between a data packet reception and a corresponding ACK packet transmission.
- $t_d$: the time required for successfully detecting an ACK from the receiver.

Because the sensitivity of RSSI depends on the RF hardware characteristics, the continuous checks of RSSI with a period of $t_r$ ensure the accurate and fast estimation of the radio channel. In order to detect the current transmission the earliest and latest RSSI checks must fall into the duration in which the sender is transmitting a packet. Moreover, in order to avoid the repeat of packet transmission from the sender when the receiver successfully receives the packet, the timing to detect an ACK packet must be less than the duration between two data packet transmissions (shown in Figure 4). Hence, the timing constraint for E-MAC can be simplified to the following condition:

$$t_a + t_d < t_i < (N - 2).t_r$$

From (1), it can be seen that E-MAC protocol does not have any constraints of packet size limit; hence, applications using small size packets (e.g. control packet) can use this protocol with high efficiency. Moreover, the reduction of complicated constraints as in [7], E-MAC also significantly reduces the computation load for microcontroller.
2.5. Optimization of transmission phase-lock

With the proposed mechanism of transmission detection and faster packet reception, E-MAC can easily learn the wake-up cycles of receiving nodes. Because sender successfully transmits a packet within the receivers wake-up window, it can find out the receivers wake-up phase. Figure 5 illustrates the procedure to lock the transmission phase in the sender after the first successful packet delivery.

Assuming $T_e$ is the point of time that a sender receives an ACK from a receiver in the first packet transmission,

$$ T_e = T_c - (t_d + t_a + 2t_p + t_i) \tag{2} $$

in which:

- $T_c$: the point of time that the sender detects ACK packet.
- $T_e$: the point of time that the sender estimates the receiver wakes up.
- $t_p$: the transmission time of a data packet (D).
- $T$: wake-up period for channel checks at receiver.

Therefore, the sender locks the transmission phase and estimates the wake-up phase of the receiver:

$$ T_e + n.T(n \in Z^+) \tag{3} $$

For the next transmissions, the sender can commence its successive transmissions to this receiver just before the receiver is expected to be awake. The transmission phase lock can be estimated so precise that the sender wants to transmit a packet to a receiver. It only sends a first packet to wake up the receiver and the next packet will be successfully received by the receiver. This helps reduce the energy consumption for transmissions and latency as well.

3. IMPLEMENTATION FOR SMART MONITORING APPLICATIONS

The E-MAC is fully implemented in the low-power devices MCU MSP430F5529 [12] and RF module CC2500 [14] and CC2538 [15] in Contiki OS. The proposed E-MAC protocol is also integrated with Thread Stack [18] to support full 6LoWPAN/IPv6 networks. The full protocol stack used for smart monitoring application is shown in the Figure 6. Table 1 shows the predefined parameters, as mentioned previously, which are used for implementation [1, 7].

In the application layer, a water quality monitoring system is designed and built with a customized sensor displayed in Figure 7. An application packet format is also used to carry the sensing water quality parameters such as pH, ORP, temperature and turbidity of the water samples.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_r$</td>
<td>0.192ms</td>
<td>$t_d$</td>
<td>0.192ms</td>
</tr>
<tr>
<td>$t_i$</td>
<td>0.5ms</td>
<td>$t_p$</td>
<td>0.16ms</td>
</tr>
<tr>
<td>$N$</td>
<td>12</td>
<td></td>
<td>1.6ms</td>
</tr>
</tbody>
</table>

4. SIMULATION AND EXPERIMENT RESULTS

4.1. Simulation

4.1.1. Operation verification of E-MAC

In this part, the improvement of E-MAC protocol is simulated with a simple networks of 6 nodes (shown in Figure 8) for analyzing the protocol operation. In this simulation all node are configured with the wake-up period ($T$) of 125ms (corresponding to the check rate of 8Hz). In the unicast transmission scenario, node 1 is the receiver, and other nodes are senders, while in broadcast transmission node 1 is the sender, and other nodes are receivers. Intendedly node 3 is located out of the signal coverage of the node 1. The radio channel activities are captured using the timeline feature of Cooja simulator at packet level.
Figure 9 explains the advantages of E-MAC in the simulated network. It can be concluded that the transmission phase-lock (Figure 9a) of node 2 using E-MAC is more accurate than ContikiMAC. After waking up, node 2 listens to channel and receives the packet correctly after the second time (indicated by an ACK from node 2 to node 1), while using ContikiMAC it only receives the successful packet after missing the first 3 packets. In addition, node 6 receives the packet which is not destined to itself; hence it does not return any ACK and enters sleep mode.

In broadcast transmission, the sender wakes up and listen channel activity to avoid collision before transmitting packet. After that it send a sequence of same packets in a wake-up period $T$ to ensure that all receiving nodes can receives at least one packet. The timeline for broadcast transmission scenario is illustrated for all MAC protocols with the wake-up period of 125ms. Different from unicast transmission, the receiver does not return an ACK to the sender and when it receives a broadcast packet, it enters the sleep mode. Moreover, the transmission phase-lock technique is not used in broadcast transmission because the packet is sent to all receivers, not only one. In Figure 9b, it can be seen that node 4 using ContikiMAC misses the short packet (sent from node 1) due to unstable short CCA strobes; however if using E-MAC, it receives this short packet successfully. This is because the E-MAC protocol does not have the constraint of minimum packet size (mentioned in Section II.D).

The completed broadcast period of all protocols are displayed in Figure 10. It is clear that in X-MAC, the idle listening time is fixed and longer than 2 times of packet transmission time to ensure that the node will get at least one broadcast packet. Therefore, there are some packet duplicates (received by node 4) as shown in the figure, which also wastes of energy. In addition, the packet size in X-MAC is not limited which is the same as in E-MAC.

Improving the longer and stable CCA checks than ContikiMAC, the transmission detection of E-MAC in the receiving node is better than ContikiMAC (shown in Figure 11a and 11b) for both cases unicast and broadcast transmission. This helps the receiving node receives the packet faster.

### 4.1.2. Performance evaluation of E-MAC

In order to investigate the performance of E-MAC protocol, a scenario of 25 sensor nodes (shown in Figure 12) is simulated in Cooja simulator [15] with the use of E-MAC protocol. Both X-MAC and ContikiMAC protocols are also used in the simulation for comparison. The energy consumption calculation is based on the powertrace tool [19].

The simulated network topology is set up with the following configuration:

- **Network depth**: 4 hops
- **The maximum number of contention nodes is 10 nodes, which are in the same area of carrier sense.**
- **Traffic load**:  
  - Low traffic intensity: inter-packet interval of 60s.  
  - Average traffic intensity: inter-packet interval of 5s.  
  - Heavy traffic intensity: inter-packet interval of 1s or 0.5s.
- **The wake-up period for all nodes**: 125ms
- **Simulated time**: 60 minutes.
4.1.3. Energy consumption

As shown in Figure 13, our proposed E-MAC optimizes the energy consumption in comparison with other protocols in all scenarios with many kinds of traffic load. With the high traffic load, E-MAC outperforms ContikiMAC approximately 13% while X-MAC gives the worst energy consumption due to the loss of transmission phase lock.

In order to investigate more on the operation of E-MAC, the energy consumption for the scenario having the data packet period of 1 second is analyzed in details illustrated in Table 2. It can be seen that although ContikiMAC spends least energy for listening radio channel, it uses more power for radio transmission and reception (22% and 11% respectively). Hence, the overall energy used by E-MAC is optimized the best thanks to the fast wake-up technique mentioned in section II.C.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>CPU active</th>
<th>CPU sleep</th>
<th>Radio Tx</th>
<th>Radio Rx</th>
<th>Radio Idle</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-MAC</td>
<td>53019.1</td>
<td>15397.7</td>
<td>83211.1</td>
<td>227905</td>
<td>8104</td>
<td>387637</td>
</tr>
<tr>
<td>ContikiMAC</td>
<td>13939.4</td>
<td>21241.3</td>
<td>60754.4</td>
<td>98914.2</td>
<td>953</td>
<td>195803</td>
</tr>
<tr>
<td>E-MAC</td>
<td>12150.6</td>
<td>21510</td>
<td>47132.2</td>
<td>87567.6</td>
<td>1563.4</td>
<td>169924</td>
</tr>
</tbody>
</table>

4.1.4. Reliability

For evaluating the communication reliability, the parameter average PDR is calculated by measuring the number of successfully received packets over the number of transmitted packets of each node in the network. The result illustrated in Figure 14 shows that, at the low traffic, all three protocols can achieve a very high PDR while both ContikiMAC and E-MAC outperform X-MAC with a PDR greater than 99%. With average traffic load, X-MAC has the worst performance (72%) while ContikiMAC and E-MAC keep a good PDR (higher 99%). However, at a high traffic load, it can be seen that E-MAC has a slightly higher PDR than ContikiMAC (approximately 3%) while X-MAC gets a poor PDR (less than 40%). Hence, E-MAC is a good protocol which should be used when the node density and the generated traffic in the network is rather high.

4.1.5. Latency

In order to compare the latency of E-MAC with other protocols, a chain topology with 10 nodes is used for simulation. All 9 nodes (node 2 to 10) send data packets to the sink node 1 with several data packet periods of 1s, 5s, and 60s.

The average latency of packets are displayed in Figure 15. It can be clearly seen that X-MAC has a...
highest packet delay because its transmission reliability is rather low; hence the number of retransmission is increased, leading to a higher delay. Moreover, E-MAC has a slightly lower delay than ContikiMAC in both single-hop delay (around 7ms) and end-to-end delay (approximately 63ms). Additionally, at the high traffic load (data packet period of 1s), the PDR of X-MAC is nearly 0, so the sink node 1 cannot receive any packets for delay measurement. The result also shows that in the case of high traffic, the packet delay of all protocols is higher than that of low traffic scenario. This is because the number of lost packets increases if all nodes generate much traffic which leads to the higher probability of receiving the high delayed packets.

4.2. Experiment

For performance evaluation in reality, a live test-bed is setup with one border-router and 20 sensing nodes using CC2538 and TelosB hardware platforms [20]. Two nodes (node 10 and 15 shown in Figure 16) in the network are equipped with the water quality monitoring device (shown in Figure 7), other nodes are configured to transmit the null packets. Each node is powered by two AA batteries with capacity of 5200 mAh. All of nodes use the same configuration of Thread stack with E-MAC integration (shown in Figure 6). To collect, visualize and analyze data packets, the tool WiSeCoMaSys [21] is modified to adapt the extract water quality parameters (e.g. pH, ORP, turbidity) from received packets.

4.2.1. Energy consumption

The energy consumption statistics are measured by collecting the energy report packets at the border-router and calculating the average energy consumption of each node. The report period can be configured using WiSeCoMaSys [21]. Figure 17 illustrates the comparison of E-MAC protocol in term of energy usage efficiency. Expectedly, the proposed E-MAC uses less energy than other protocols in all kinds of traffic level (low or high traffic). With the high traffic configuration, X-MAC losses most of the packets; hence there are not enough statistics for this protocols energy calculation. Additionally, ContikiMAC uses more energy than E-MAC does (approximately 14% for all traffic load levels). Again, this result confirms the conclusions in the simulation section.

4.2.2. Reliability

At the border-router, the average PDR of the network is also measured by counting the number of received packets in many cases of data packet periods. As shown in Figure 18, X-MAC shows the worst PDR in all scenarios, while at the very high traffic level (with data packet interval less than 1 second), E-MAC outperforms ContikiMAC approximately 6%. If the data packet period is configured with a value greater than one seconds, both ContikiMAC and E-MAC get the same average PDR (greater than 99%).

5. Conclusion

With the proposed E-MAC protocol, it is believed that this protocol can be a suitable candidate for many applications, especially in monitoring applications (e.g. habitat monitoring, smart control system) where all sensor nodes need a long lifetime and stable communication. The experiment measurement also shows that E-MAC can reach lower power consumption in comparison with other MAC protocols while keeping a good PDR and low latency. In the future, E-MAC will be improved with the feature of packet segmentation so that the communication between nodes can support any different large packet sizes. Another issue is to port E-MAC to the popular TinyOS operating system [22].
Acknowledgment

This research is funded by the Vietnam National University-Ho Chi Minh City (VNU-HCMC) within the Science Research Project under grant number C2016-20-03.

References


Vo Que Son was born in Quang Ngai, Vietnam. He received the B.E. degree in Electrical and Electronics Engineering from Ho Chi Minh City University of Technology, Vietnam in 2003. Two years later, he got the M.E. degree in Telecommunications Engineering from the same university. He received the Ph.D degree in Electrical Engineering from University of Bremen, Germany in 2011. Since 2004, Dr. Son has been a faculty member of Department of Electronic Engineering at Ho Chi Minh City University of Technology. His research interests lie in Mobile and Wireless Communications, Wireless Sensor Networks, Internet of Things, Autonomous/M2M communications. He is an author of several journals and tens of conference papers.

Tran Truong Son was born in Quang Binh, Vietnam. He received the B.E. degree in Electrical and Electronics Engineering from Ho Chi Minh City University of Technology, Vietnam in 2013. Three years later, he got the M.E. degree in Telecommunications Engineering from the same university. Currently, he is an engineer of MobiFone, a major Vietnamese mobile network operator. His research interests focuses on Radio Access Network, Metropolitan Area Network, Wireless Sensor Networks and Internet of Things.