# MAC PROTOCOLS FOR IMPROVING PERFORMANCE OF WIRELESS SENSOR NETWORKS

Neha Vishwakarma *

Prof. Vineet Richhariya **

## ABSTRACT

*The MAC schedule should be chosen to maximize the lifetime of the network, which includes reducing contention. Utilizing this information at the MAC level is made possible. MAC protocols employ identical schedules for both unicast and broadcast packet transmissions or, when impossible, simply modify their "unicast schedule" to work with broadcast packets. For instance, IEEE 802.11 cannot perform an RTS / CTS handshake for broadcast packets, and thus only utilizes CSMA for broadcast packets, regardless of the impact on lifetime or contention. In this paper, we propose adapting the MAC schedule to node and network conditions to improve performance under a wide range of conditions and for both unicast and broadcast packets. The MAC schedule should be chosen to maximize the lifetime of the network, which includes reducing contention. MAC schedule to node and network conditions to improve performance under a wide range of conditions and for both unicast and broadcast packets the "transmit / receive schedule" to synchronize nodes on a slowly changing path so that throughput and delay are further reduced, at no cost of overhead in most cases. the MAC schedule provides to increase in lifetime for different traffic scenarios. the transmit / receive schedule to automatically synchronize the nodes can reduce packet delivery delays, providing an efficiency and throughput will increases.*

*Keywords- Wireless sensor networks (WSNs), Transmit, Receive Schedule, MAC Schedule , Efficiency and Throughput, IEEE 802.11, unicast and broadcast packet, network.*

* M-Tech (CSE), LNCT, Bhopal,

** Professor, Department of Information Technology, Bhopal, Madhya Pradesh, India.

# I. INTRODUCTION

The Wireless sensor networks (WSNs) have been used for many applications, including environmental monitoring, health monitoring, security and surveillance [1]. These different applications for WSNs have vastly different bandwidth requirements. Take, for example, visual sensor networks (VSNs) for surveillance or health monitoring. wireless sensor networks (WSN) are becoming increasingly complex, and they require the network to maintain a satisfactory level of operation for extended periods of time. Consequently, sensor networks have to make the best possible use of their initial energy resources, specifically by constantly adapting their protocols to the changing conditions in the network. Both protocol-specific and cross-layer schemes have offered a plethora of energy reducing techniques. In particular, there are several protocols that focus on reducing energy at the data link / MAC layer, which constitutes the scope of this work. In this paper, we investigate how to keep the radio in its increase efficiency sleep mode for as long as possible. In this paper discusses MAC schedules to adopt the most throughput efficient pattern of packet transmissions and receptions. Because different areas in the network experience different and changing loads of traffic, the MAC protocol should utilize the schedule most economical for the local conditions. We also propose to synchronize nodes so as to reduce transmission time and thus throughput decreases and packet delivery delays.

Here technique controls the inter-listening time to conditions in the network and is exposed in [2]. As new sensor network platforms have appeared on the market, a simple observation was made that idle listening, far from being negligible, was a major source of energy and time consumption [3]. the sending node occupies the medium for long intervals to signal its imminent packet transmission. Receiving nodes are thus allowed to sleep for at most the duration of this preamble and they must stay awake when they sense a busy medium until the packet transfer is complete. (although many of our results can be transposed to other MAC protocols), and we define ) MAC schedule" as the pattern of packet transmissions occurring within the interval.

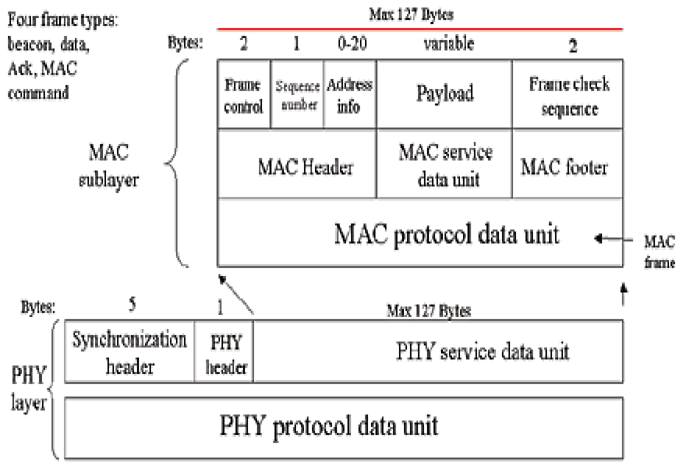


**Figure 1 shows the construction of the data frame, also called a *data packet*.**

Changes in radios have forced researchers to abandon BMAC and a few other LPL protocols in some cases: although it paved the way to new MAC protocols, B-MAC, which uses a variable-length preamble to signal the impending packet transmission, can no longer be implemented as proposed on the new IEEE 802.15.4 compliant platforms because this standard has a fixed preamble length of only a few bytes.

We propose to modify the MAC schedule of X-MAC by repeating the data packet and waiting for ACK frames between transmissions. A received ACK signifies that the data packet has been correctly received and stops the transmission flow of data packets. This renders the MAC protocol immune to false positive packet receptions. For broadcast packets, the flow of data packets is still interleaved with periods of listening for acknowledgments. Consequently, multiple receivers of the same packet may wake up, stay in Rx mode until the full reception of a packet, and go back to sleep. Figure 2 illustrates the timeline for this modified X-MAC protocol, called MX-MAC. We assume such a target radio, and make design and research decisions accordingly—thus B-MAC is not included in our work. After the introduction of new radios, researchers introduced new LPL and PS protocols: X-MAC [4], C-MAC [5], WiseMAC [6], CSMA-MPS [7] and SpeckMac [8] are among the most popular contributions. These protocols are based on repeating either the data packet itself (SpeckMAC and CSMA-MPS), or an advertisement packet (X-MAC / CMAC), in place of a long preamble. The details of the transmission schedules (the "MAC schedules") are given in Figure 2.

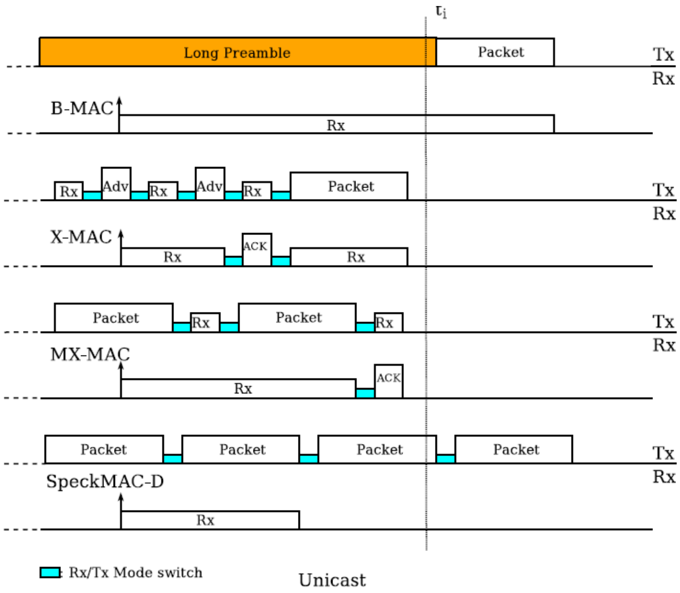


**Figure 2 MAC schedule for B-MAC, X-MAC, MX-MAC, and SpeckMAC.**

In the initial part of this work, we prove that while the MAC protocols generally throughput consumption without resorting to explicit exchange of active / inactive schedules between nodes, low duty cycles drastically favor receiving nodes over mostly sending nodes and induce higher delays and contention. As Figure 2 shows, only one data packet can be transmitted per ti cycle, which can cause a packet to experience high delay over several hops, and the network to deliver small data rates. Concern for delay may force network designers to select a high duty cycle that would limit energy savings. We address this problem in the second part of our work by synchronizing the transmitting / receiving schedules of nodes on a slowly-changing routing tree.

• We propose switching MAC schedules from a pool of MAC protocols at the transmitter to minimize energy consumption based on parameters such as packet size, whether the packet is broadcast or unicast, and the estimated ratio of transmit to receive packets in the local neighborhood. The protocols are "compatible" because they are interchangeable: the receiver does not need to know what specific schedule is being used, it simply wakes up and senses the channel every ti seconds and sends an ACK frame when required by the received packet.

• We propose to synchronize nodes along a slowly changing routing path so as to minimize error and packet delay, without explicit scheduling between nodes or overhead of any sort. For unicast packets, the sender stops its stream of advertisement (X-MAC / C-MAC) or data (MX-MAC) packets after receiving an acknowledgement frame. Sender and receiver can then be synchronized to wake-up sequentially within a short interval. Conversely, SpeckMAC, which cannot be interrupted, needs explicit notification within nodes to synchronize.

## II. LITERATURE SURVEY

A MAC protocol that reduces the preamble length before sending a data packet by exchanging wake-up schedules between neighbors. However, WiseMAC (like B-MAC) cannot be implemented on 802.15.4 radios. It also requires fine time synchronization between nodes, and at a cost that may be difficult to quantify. For these reasons, it was not included as such in our study. Moreover, other work [9] shows that implicit synchronization can be achieved between nodes running some of the protocols studied here, reducing the need for synchronization overhead.

The authors thoroughly compare B-MAC to SMAC [10] and T-MAC [11]. To curb limitations imposed on the receiving node to stay awake for the time of the preamble, Polastre et al. propose sending packets with half-sized preambles. Post-B-MAC protocols include X-MAC [12] and SpeckMAC-D [13]. Both protocols are of the channel-probing family and tried to improve the scheme presented by B-MAC. Further explanation of these protocols is provided in Section 3. Although more recent, C-MAC [14] uses the same schedule as X-MAC and is therefore included in our work under the same principles that govern X-MAC.

In [13], Wong and Arvind also propose SpeckMAC-B, which is compared, along with SpeckMAC-D, to B-MAC. The SpeckMAC protocol family is intended for miniature motes called specks. SpeckMAC-B stands for Back off and replaces the long preamble with a sequence of wake up packets containing the destination target and the time when the data packet will be sent. This allows receiving nodes to sleep for the remainder of ti and activate just in time for data reception. However, this scheduling supposes fine time synchronization between nodes, which we do not assume in this work.

## III. PROPOSED TECHNIQUE

### *A. Reconstruction Model*

We accurately evaluate the lifetime of a mote by measuring the energy consumed under various basic operations using a fast data acquisition board. We measured the throughput and time spent probing the medium, starting a transmission, sending one packet and switching the radio back to TX mode, stopping a transmission after a successful and failed (only for X-MAC and MX-MAC schedules) transmission, and receiving a packet. Every scenario is then reconstructed with Matlab by adding the energy expended during each operation. Moreover, our measurements allow us to determine the radio switch time Selecting a MAC protocol supposes a compromise between excellent performance under certain circumstances (hoped to be the common case), and suboptimal operation otherwise. Various protocols may perform differently according to the broadcast / unicast nature of the packets, We propose creating a pool of MAC protocols that are compatible with one another: while the sender may decide which schedule to follow based on the parameters mentioned above, the receiver need not be informed of the changes in MAC Protocals. For instance, a sender choosing a certain MAC Protocals may expect an ACK frame between packet transmissions; it will thus stay in receiving mode for a given time before it returns to transmitting mode. At the other end of the communication, a receiver simply wakes up periodically, and occasionally receives packets. If a received packet is marked with an acknowledgment request, it immediately sends an ACK frame. Switching between interchangeable MAC Protocals guarantees that gains in energy and latency are achieved without any overhead save the computation required to determine the best Protocals to use.

This solution proves inexpensive in terms of computation power during runtime. The threshold values dictating a change in the MAC schedule can be established before deployment using simulation and implementation results. As we will show, in accuracies in the estimates of current node, network and application conditions do not have a major impact on the performance of MAC. Existing MAC protocols were included as part of the pool of compatible MAC schedules: X-MAC [12] and SpeckMAC-D [13], which were introduced around the same time. However, we also add a novel MAC schedule based on a modified version of X-MAC.

***B. Protocol Design Choices***

We tried to optimize as many aspects of the MAC Protocals as possible: the time separating two clear channel assessments (CCA) as well as the number of CCAs when sensing the medium, the number of CCAs before a packet transmission, the behavior of a node when it detects another ongoing transmission (what a sender should do when hearing another stream of packets during its switch to RX mode after every frame), etc. Since our goal is only to compare MAC protocols without bias toward one Protocals, we endeavored to optimize the behavior of MAC protocols. Because all MAC schedules are meant to be compatible, they will be implemented by the same Matlab code. Consequently, all three protocols have the same essential parameters such as the number of CCAs and the time separation between them. For packet sizes close to their maximum value (128 B), we found the radio to “jam” under SpeckMAC: the radio would issue RXFIFO overflows because the FIFO was filled before it could be read, and hence the packet delivery ratios in this case dropped significantly.

### *C. MAC Schedule Compatibility*

Through design choices, we allowed the three MAC protocols to be compatible. More importantly, the basic principle behind schedule compatibility is that a receiver does not need to know the ongoing schedule, and simply ACKs packets that request it. For MXMAC packets and for X-MAC advertisements, the acknowledgement request field must be set to one. If no ACK is requested, the receiver simply turns off after the packet has been received.

### *D. Lifetime for Unicast Packets:*

The lifetimes advertisement packet size is 40 B (not the original X-MAC's 11 B), which increases the chance of being heard during a transmission, and thus saves retransmissions. At the same time, an increase in packet size increases the lost in packet. This is because the radio transmits for ti s or until interrupted, whatever the packet size. The advantages of MAC are reduced further when the data packet size reaches that of the advertisement size because the advertisement packet is no longer easier to hear than the data packet. the packets smaller than 40 B, and for cases when the node is mostly sending, X-MAC allows the node to increase its lifetime. In other cases, MX-MAC leads to a longer lifetime. This is possible because while the receiver does not get to pick the MAC schedule, the sender can select the appropriate MAC given current network and neighbor conditions. The receiver does not need to be informed of any hanges in MAC scheduling. Based on the packets received.

### *E. Lifetime for Broadcast Packets*

Contrary to unicast packets, one MAC schedule consistently spares the throughput of the node, over the range of packet sizes. the broadcast small gains in lifetime, and it will greatly improve packet delivery.

## IV. TEST RESULTS AND ANALYSIS

Matlab implementation will demonstrate the feasibility of compatible MAC schedules and the benefit of switching between them. No extra overhead was required, save the memory required. MAC schedule may be adapted to conditions in the network to increase lifetime or throughput. On top of the gains obtained through MAC schedule, we also propose a simple approach to synchronizing nodes on a temporarily-fixed path for the sub-family of protocols. Through analysis, we proved that the path is automatically synchronized after n = h packets have been sent from node 0 (the farthest) to node n. In other words, the requirement to have a fixed path is a weak one since it needs to be constant for only h packets.

Synchronization of transmit / receive schedules has several benefits: it drastically reduces the packet delay, and it reduces the energy use at every node by a factor of about ti/2tS , removing the limit standing in the way of lower duty cycles. In addition, we proposed several strategies to increase the packet rate and further reduce the packet delay. Pipelining packets over synchronized paths doubles the packet rate. Urgent packets are delivered almost immediately, taking the delay from $t_S + (n-1)(t_i + t_S) + t_{Rx}$ to $nt_S + t_{Rx}$. Although MAC and node synchronization may be implemented without the benefit of the other, their combined impact on node lifetime and packet delivery delay exceeds that of each approach independently. This is because MAC may select the most reliable MAC schedule, which in turns greatly facilitates node synchronization.

## V. CONCLUSION

In this paper, we propose a MAC schedule to node and network conditions to improve performance under a wide range of conditions and for both unicast and broadcast packets. the number of parameters and metrics to switch MAC schedules. The MAC schedule should be chosen to maximize the lifetime of the network, which includes reducing contention. MAC can significantly increase per-node lifetime and that node synchronization is both possible and practical.

Existing MAC protocols employ identical schedules for both unicast and broadcast packet transmissions or, when impossible, simply modify their “unicast schedule” to work with broadcast packets. For instance, IEEE 802.11 cannot perform an RTS / CTS handshake for broadcast packets, and thus only utilizes CSMA for broadcast packets, regardless of the impact on lifetime or contention.

the improvements on the node lifetime and packet delays require no overhead or cost in most WSN cases: nodes do not need to exchange On / Off schedules with their neighbors, and in the unidirectional case, no explicit synchronization phase or messages are required. the nodes organize themselves automatically. We will implement and test in matlab.

## VI. REFERENCE


1. D. Cassioli, M. Z. Win, and A. F. Molisch, “A statistical model for the UWB indoor channel,” *Proc. Vehicular Technology Conference(VTC)*, vol. 2, pp. 1159–1163, 2001.
2. C. J. Merlin and W. B. Heinzelman, “Duty cycle control for low power listening MAC protocols,” in *Proceedings* 5th *IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS’08)*, Sep. 2008.
3. J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2*nd *ACM Conference on Embedded Networked Sensor Systems (SenSys’04)*, Nov. 2004, pp. 95–107.

4. M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4*th *Embedded Networked Sensor Systems (Sen- Sys'06)*, Nov. 2006, pp. 307–320.
5. S. Liu, K.-W. Fan, and P. Sinha, "CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," in *Proceedings of the Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'07)*, Jun. 2007.
6. A. El-Hoiydi and J. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, Jul. 2004.
7. S. Mahlknecht and M. B¨ock, "CSMA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks," in *Proceedings of the IEEE International Workshop on Factory Communication Systems*, Sep. 2004.
8. K.-J.Wong and D. Arvind, "SpeckMAC: Low-power decentralised MAC protocol low data rate transmissions in Specknets," in *Proceedings 2*nd *IEEE International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN'06)*, May 2006.
9. Merlin, C.J., Heinzelman, W.B.: Node synchronization for minimizing delay and energy consumption in low-power-listening MAC protocols. In: Proc. MASS '08. (September 2008)
10. Wei, Y., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proc. INFOCOM '02. (June 2002)
11. Van Dam, T., Langendoen, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proc. SenSys '03. (October 2003)
12. Buettner, M., Yee, G.V., Anderson, E., Han, R.: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proc. SenSys '06. (November 2006) 307–320
13. Wong, K.J., Arvind, D.: SpeckMAC: Low-power decentralised MAC protocol low data rate transmissions in specknets. In: Proc. IEEE International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN '06). (May 2006)
14. Liu, S., Fan, K.W., Sinha, P.: CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. In: Proc. SECON '07. (June 2007).