© December 2024 | IJIRT | Volume 11 Issue 7 | ISSN: 2349-6002

# A Multi-Layer Strategy for Reducing Interference and Latency in Medium Access of Wireless Sensor Networks

BAMULI SWAPNA[1], ALLA RAJU[2]
*[1,2] Assistant Professor Vaagdevi Degree & PG College, Hanamkonda, Telangana-506001*

*Abstract:* **In low-power wireless sensor networks, MAC protocols typically use periodic sleep/wake schedules to minimize idle listening time. While this approach is simple and efficient, it leads to higher end-to-end latency and reduced throughput. In contrast, earlier CSMA/CA-based MAC protocols aimed to reduce inter-node interference but at the expense of increased latency and lower network capacity. In this paper, we introduce IAMAC, a CSMA/CA sleep/wake MAC protocol that reduces inter-node interference and minimizes per-hop delay through cross-layer interactions with the network layer. We also demonstrate that IAMAC can be integrated into the SP architecture for effective inter-layer interactions. Through extensive simulations, we evaluate IAMAC's performance across various metrics. The results confirm that IAMAC reduces energy consumption per node and extends network lifetime compared to S-MAC and Adaptive S-MAC, while also offering lower latency than S-MAC. Our evaluations consider IAMAC with two error recovery methods—ARQ and Seda—and show that using Seda as the error recovery mechanism improves throughput and lifetime over ARQ.**

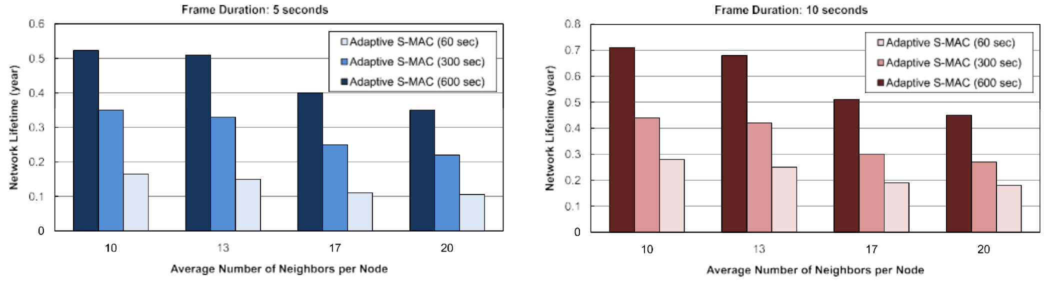
*Keywords:* **Wireless Sensor Networks, Medium Access Control (MAC), IAMAC, Tree-Structured Routing, Cross-Layer Enhancement, Interference Management.**

## 1. INTRODUCTION

Wireless communication in wireless sensor networks has been explored in highlighting the irregularity and unreliability of low-power wireless links. These studies identify three distinct reception regions in a wireless link: connected, transitional, and disconnected. Most links to neighboring nodes fall into the transitional region, which experiences significant fluctuations in link quality due to factors like environmental noise and inter-node interference. As a result, many suboptimal links may be selected by routing algorithms. Although link estimation can help choose the best next-hop neighbor, concurrent transmissions from neighboring nodes can cause immediate fluctuations in link quality, leading to inter-node interference. This interference, along with a high packet corruption rate, increases energy consumption per node, which contradicts the goal of extending the lifetime of small, low-power sensor nodes. Current routing algorithms and MAC collision avoidance methods fail to fully address the effects of inter-node interference. In the MAC layer, to avoid collisions in S-MAC nodes that overhear control packets (i.e., RTS and CTS) are prevented from transmitting. However, because of the multi-hop nature of packet transmission in wireless sensor networks, this mechanism leads to significant end-to-end latency. To address this issue in S-MAC, Adaptive S-MAC introduces an adaptive node activation mechanism based on the estimated transmission duration between neighboring nodes.

## 2. DRAWBACKS OF THE ADAPTIVE S-MAC PROTOCOL

S-MAC and Adaptive S-MAC are sleep/wake MAC protocols designed for wireless sensor networks, with energy efficiency and latency being crucial evaluation criteria. To minimize packet collisions, S-MAC prevents nodes that overhear control packets (such as RTS and CTS) from transmitting. Although this reduces energy consumption, it also leads to high delays. To tackle this problem, Adaptive S-MAC uses adaptive node activation, adjusting based on the estimated transmission duration between neighboring nodes. While Adaptive S-MAC reduces latency compared to S-MAC, it still has two major limitations: First, during communication between two nodes, neighboring nodes must overhear RTS and/or CTS packets to learn the approximate duration of the communication. This allows them to wake up earlier than their scheduled time to transmit data. While this can lower delay by enabling data to traverse multiple hops within a frame (defined as a full cycle of listening and sleeping), the highly variable link quality in low-power wireless sensor networks causes inaccurate transmission duration estimates. As a result, idle listening time and energy consumption per node increase. The additional energy consumption depends on factors like the average number of neighboring nodes, radio type, and environmental conditions. Among these, the number of neighboring nodes significantly limits the protocol's scalability. As more nodes overhear control packets and wake up adaptively, network energy consumption rises, influenced by radio switching and channel sampling costs. Additionally, when environmental conditions cause more instability in radio links, neighboring nodes face longer idle listening times due to inaccurate wake-up schedules.



(a) (b)

Figure 1. Variations of the Adaptive S-MAC’s lifetime versus the average number of neighbors per node. The value in each parenthesis indicates the packet generation interval ateach node.

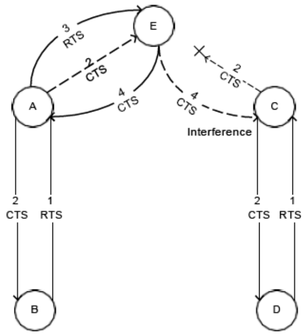
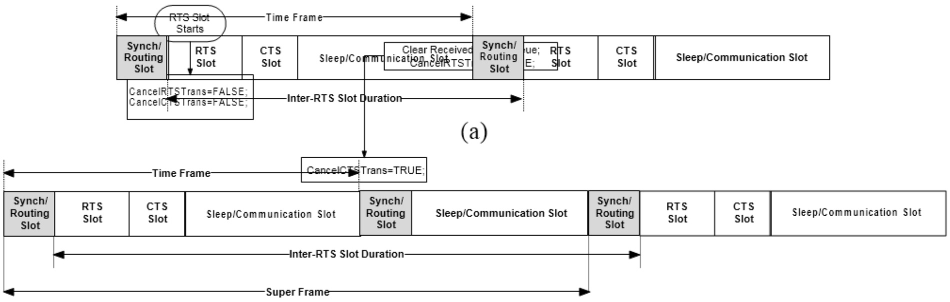


Figure 2. A sample scenario for interference in Adaptive S-MAC. The number on each arrow shows the time sequence. At time 1, node B and node D send RTS to node A and node C, respectively. When node A accepts data reception by sending CTS to node B, nodeE overhears this packet and captures the communication duration between node A and node B. Moreover, node E cannot overhear the CTS packet transmitted from node C to node D. Therefore, after the communication between node A and node B finishes, node E wakes upand interferes with data reception at node C.Adaptive S-MAC we evaluated the lifetime of this protocol against the average number of neighbors per node in Figure 1. Our general simulation settings for the simulations of this paperare described in Section 4. According to Figure 1, Adaptive S-MAC is not scalable, i.e., as the average number of neighbors per node increases, the network lifetime decreases. This is the effect of increase in the number of the nodes, which adaptively wake up according to the communication duration between two neighboring nodes.

## 2. THE PROPOSED MAC PROTOCOL

In this section we introduce our proposed cross-layer MAC protocol. However, before proceeding to the MAC protocol description, we first introduce the routing algorithm.



(a)

(b)

Figure 3. Time Frame and Super Frame structures. (a): If the duration between two consecutive RTS Slots is less than 12 seconds Time Frame structure can be used. (b): If the duration between two consecutive RTS Slots is more than 12 seconds Super Frame structuremust be applied. When using Super Frame structure,

Time Frame duration should not exceed12 seconds.

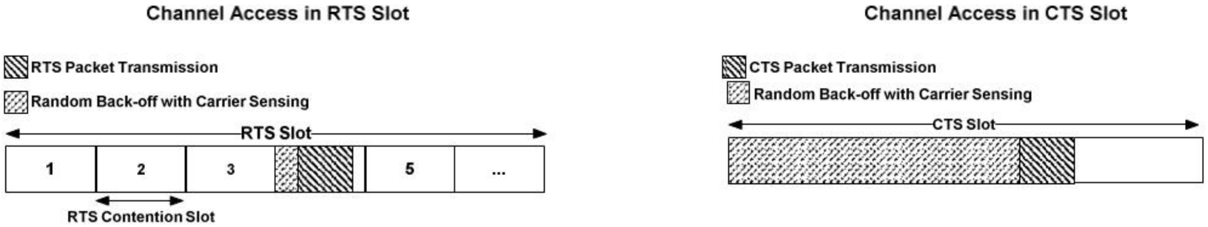


Figure 4. Channel Access Mechanisms During RTS Slot and CTS Slot

In this way, the probability of concurrent transmission of CTS packets will be very small due to the trivial signal propagation delay in wireless sensor networks.

IAMAC provides two different algorithms for RTS Slot and CTS Slot. Before explaining these algorithms, there are some points worth clarifying here. First, each node, in addition to the packet queue, must also include another queue to keep its received RTS packets during a Time/Super Frame; this queue is called *Received RTSs Queue*. Second, to prevent inter-node interference and control node operations, two Boolean control variables are defined: *CancelRTSTrans* and *CancelCTSTrans*. Third, in the proposed algorithms, node deactivation forces the node to go into sleep mode immediately.Figure 5 demonstrates the flowchart of RTS Slot's algorithm. Because evaluating all the possible scenarios regarding to this algorithm is infeasible, we provide a somewhat simple scenario to clarify the operation of this algorithm. This scenario is depicted in Figure 6. The time below each step indicates the time progress as RTS packets are being received. Since eachRTS packet can be received in a RTS Contention Slot, each time step corresponds to a RTS Contention Slot (notice that these time steps are not necessarily consecutive RTS Contention Slots). At Time 1, node *A* sends its RTS packet to node *B*. Upon reception of this packet, node *B*adds it to its Received RTSs Queue. When node *B* overhears a RTS transmission from node *E* to node *C* at Time 2, it conceives that replying to node *A* may result in inter-node interference. In addition, since the destination address of this overheard packet is the same as the parent addressof node *B*, node *B* deletes the received RTS packet from its queue and changes its control variables to a new state. These new values for control variables allow node *B* to be a data transmitter (allowing it to send a RTS packet to its parent). Transmitting RTS packet from node*B* to node *C* at Time 3 does not change the state variables of node *E* because the overheard packet's destination address is the same as the parent address of node *E*. However, overhearing.

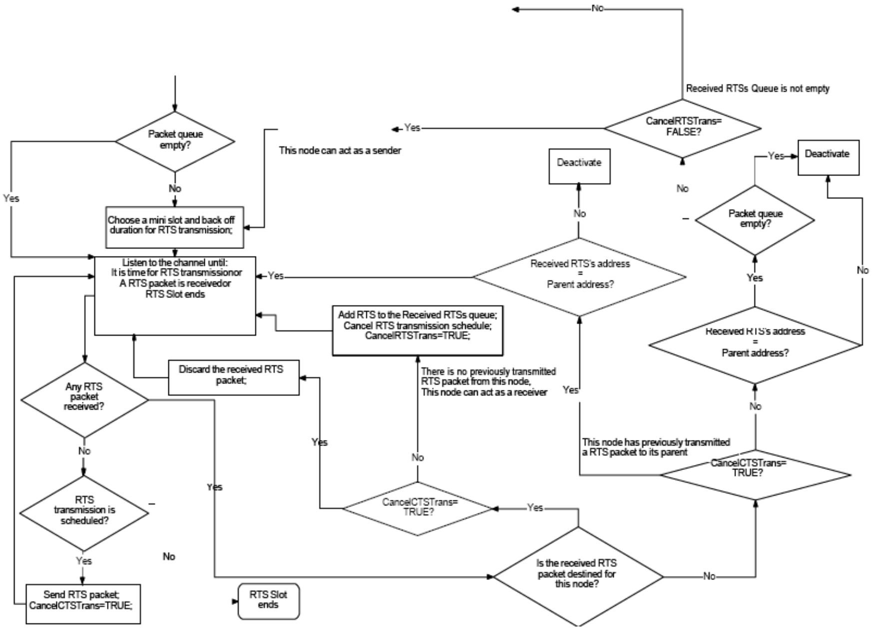


Figure 5. Flowchart of RTS Slot's Algorithm

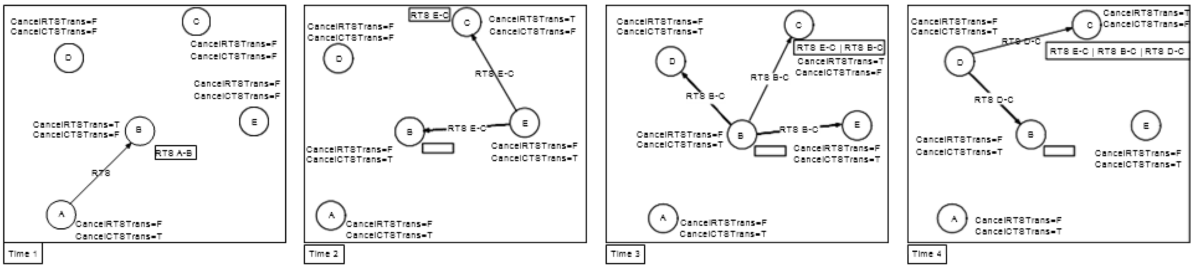


Figure 6. A Sample Scenario for RTS Slot

this RTS packet at node *D* changes its control variables to a new state since it can act only as a sender. At Time 4, node *D* sends its RTS packet to node *C*. Notice that overhearing this packet at node *B* does not change its control variables. At the end of the RTS Slot, node *C* includes RTS packets from *E*, *B*, and *D*; so it must respond to these packets during the CTS Slot. The respond mechanism can be performed in two ways: (1) using a single broadcast or (2) a CTS packet for each received RTS packet. Though using a single CTS packet is more energy efficient, incorrect packet reception at each node prevents that node from receiving its schedule for data transmission to its parent. In contrast, by transmitting multiple CTS packets we provide higher packet reception probability at the child nodes. In this paper, we use the maximum number of CTS transmissions, i.e., a CTS packet will be transmitted for each received RTS packet. The second algorithm is the CTS Slot's algorithm, which acts as a complementary algorithm for RTS Slot's algorithm. Since the CTS Slot's algorithm is not complex and in order to provide some implementation details we provide it in the form of pseudo code. The corresponding algorithm can be seen in Algorithm 1.

Algorithm 1. Algorithm for CTS Slot

```
/*If the Received RTSs Queue is not empty and this node is allowed to send CTSpacket:*/
If (ReceivedRTSsQueue.Length!=0)
    Choose a random time for CTS transmission;

While (CTS Slot is not finished)
{
    //When a CTS packet is received:
    If (a new packet is received)
        Pkt=Arrived Packet;

    If ( (CTS timer is reached) && (channel idle) )
        Send CTS packet;
        //This can be a single or multiple consecutive CTS transmissions

    //Overhearing a CTS packet, cancel CTS transmission:
    If ( (Pkt.RecAddress!=MyAddress) || (channel busy) )
        Deactivate;
        /*Due to link asymmetry and RTS packet corruption, we may receive a CTS packet that is not destined for this node. Therefore, in order to avoid interference, this node must be deactivated*/

    /*If this node receives a CTS packet, it is allowed to transfer its data in Sleep/Communication Slot:*/
    If (Pkt.RecAddress==MyAddress)
        Prepare for data transmission;
}
```

The RTS Slot duration depends on the average number of children per node. Furthermore, since the RTS Slot duration should be equal for all the nodes, scalability problems may appear. In order to remedy this problem, we can limit the maximum number of children per node. To this aim, when a node wants to select its parent, it also considers the number of

Table 1. Default Simulation Settings

| Radio | | | |
|---|---|---|---|
| Modulation | FSK | Encoding | NRZ |
| Output Power | 0 dBm | Frame | 45 bytes |
| Transmission Medium | | | |
| Path Loss Exponent | 4 | $PL_{D0}$ | 55 dBm |

| | | | |
|---|---|---|---|
| Noise Floor | -105 dBm | $D_0$ | 1 m |
| Other Parameters | | | |
| Number of Nodes | 200 | Area | 100×100 m$^2$ |

Table 2. Detailed Parameters for ARQ and Seda

| Parameter | Symbol | Value |
|---|---|---|
| Maximum Packets per Frame | $MPF$ | Variable |
| Payload Length | $L_l$ | 29 |
| Physical and MAC Headers Length | $L_{phy\_mac}$ | 16 |
| Packet Length | $L_P$ | 29+16 |
| Block Overhead | $L_{BO}$ | 2 |
| Block Length | $L_B$ | 29+2 |
| ACK Packet Length | $L_{ack}$ | 23 |
| Radio Speed (bps) | $S_R$ | 19200 |
| Bit Error Rate | $BER$ | Variable |
| Recovery Frame Overhead (byte) | $RF_{OV}$ | 5 |
| Sleep Duration (second) | $D_S$ | Variable |

children that its neighboring nodes currently have. Therefore, each node looks for a qualified node in terms of cost and number of children, and then selects that node as its parent.

## 4. EVALUATION

For precise evaluation of sensor network protocols, accurate modeling of wireless channel is of great importance. Accordingly, we implemented the link layer model from USC in OMNeT++ framework. Then, IAMAC, S-MAC, Adaptive S-MAC, and spanning tree routing algorithm were implemented in separate modules. Table 1 represents our general simulation settings similar to the characteristics of MICA2 motes. The sink node is positioned at the middleof top edge. Table 2 provides more details regarding the data link layer parameters. Energy consumptions of radio and sensor operations are provided in [9]. In our evaluations, we may change some of these parameters with notification.

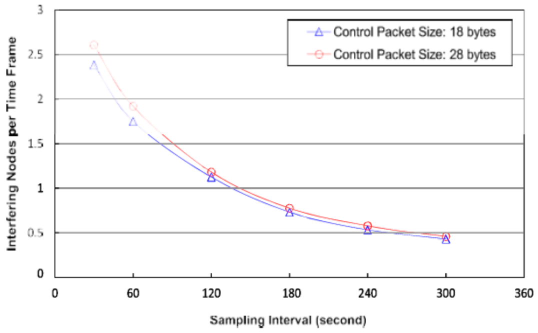


Figure 7. Interfering nodes per Time Frame versus sampling interval. (Time FrameDuration=1 sec.)

In order to measure the maximum network throughput, we forced each node to sample the environment as fast as it can and transmit its data packets with maximum capacity. Figure 10 shows the throughput of IAMAC in combination with Seda and ARQ. According to this figure, IAMAC with Seda achieves higher throughput than ARQ, which also confirms our analytical results. In this figure, notice the rise and fall of the network throughput that is similar to the behavior observed in Figure 8.

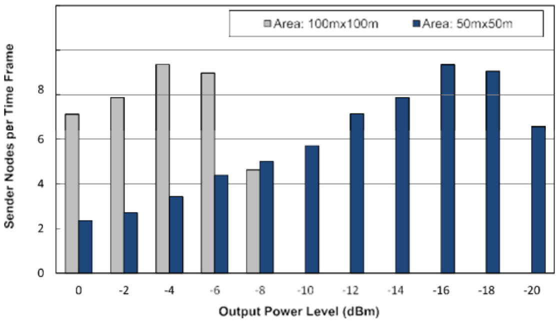


Figure 8. Average number of sender nodes per Time Frame. Each network density corresponds to an optimal output power level, which trades off between radio interference level and number of children per node. (Time Frame duration=1 sec, sampling interval=60 sec.)

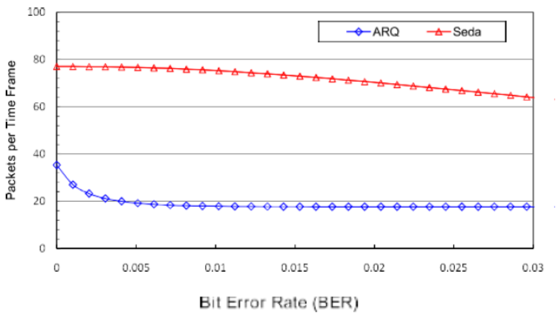


Figure 9. Number of data packets per Time Frame, considering one retransmission for everycorrupted packet/block. (Time Frame duration=1 sec.)

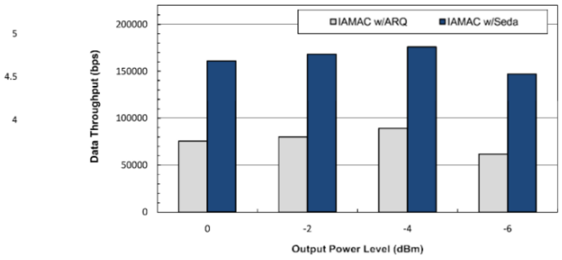


Figure 10. Effect of output power level on network throughput. (Time Frame duration=1 sec, sampling interval=1.1 sec.)deactivations cannot result in noticeable increase of lifetime. This behavior is also visible in Figure 12, in which the average duty cycle of the nodes is demonstrated.

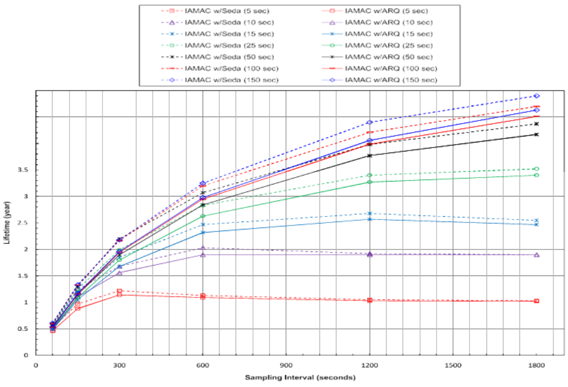


Figure 11. IAMAC's network lifetime as a function of sampling interval. The value in eachparenthesis indicates the Time/Super Frame duration. As the sampling interval increases, the lifetime also increases because less time is spent on transmission and reception of data packets. Increasing Time/Super Frame duration also increases lifetime. This is due to the lessoverhead of active slots (i.e., Synch/Routing Slot, RTS Slot, and CTS Slot), compared to thewhole Time/Super Frame duration. Also, Seda can improve the lifetime of IAMAC and thisimprovement is more evident for long sampling intervals and lengthy Time/Super Frame durations. When number of transmitted data packets in each Time/Super Frame is high, Sedacan benefit from its low packet corruption rate and efficient error recovery.Figure 13 demonstrates the lifetime of IAMAC against S-MAC and Adaptive S-MAC. It is evident that IAMAC provides significant increase in lifetime compared to Adaptive S-MAC. As discussed in Section 2, the lower lifetime of Adaptive S-MAC is mainly due to its adaptive listening mechanism. Even though with equal Time Frame durations IAMAC provides lower lifetime against S-MAC, it will be shown in Section 4.4 that IAMAC obtains higher performance than S-MAC in terms of lifetime and delay.

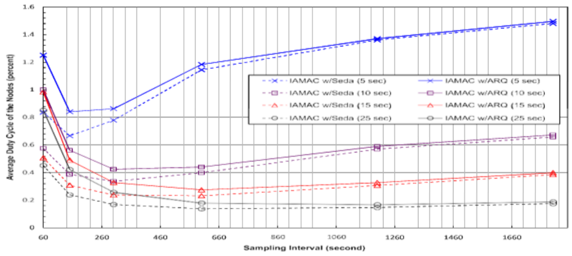


Figure 12. Variations of duty cycle against sampling interval. Notice the fall and rise of eachduty cycle around a specific sampling interval. These minimum values for average duty cycle appear as the result of trade off between node active time, number of sequential transmissions per Time/Super Frame, and number of deactivated nodes. For long Time/Super Frame durations, the average duty cycle will be inherently low and this behavior is less evident.

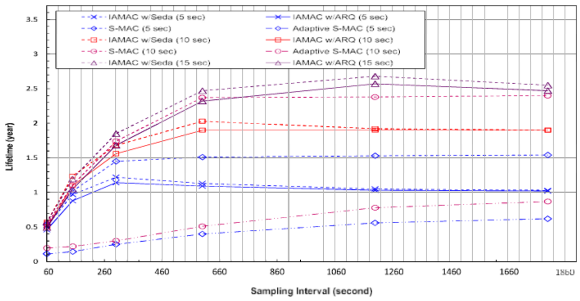


Figure 13. Network lifetime of IAMAC, S-MAC, and Adaptive S-MAC versus sampling interval. The value in each parenthesis demonstrates the Time/Super Frame duration for IAMAC and frame duration for S-MAC and Adaptive S-MAC.

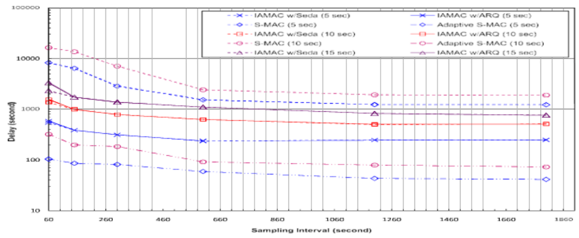
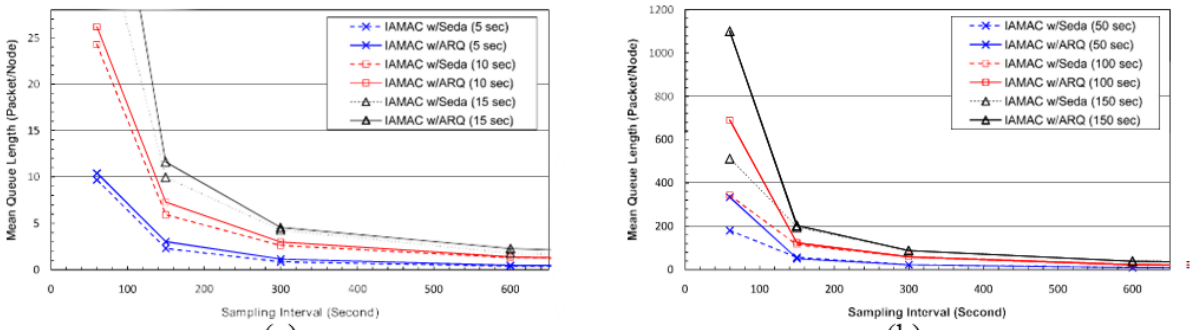


Figure 14: End-to-end delay of IAMAC, S-MAC, and Adaptive S-MAC. The value in each parenthesis demonstrates the Time/Super Frame duration for IAMAC and frame duration forS-MAC and Adaptive S-MAC.

## 5. ARCHITECTURAL ISSUES

Although increasing inter-layer interactions in cross-layer optimization provides more opportunities for performance optimization, however, the effects of these interactions must be considered carefully. Establishing connections and interactions between different protocols maydestroy system modularity and impede the understandability and optimization of the protocols [11][12][19]. To this aim, SP architecture [13] tries to provide richer inter-layer interactions while it also preserves modularity. In this architecture, through the SP abstract layer the upper and lower layers can communicate with each other. On the other hand, as we have seen before,IAMAC is based on the interactions of MAC and network layer. Accordingly, IAMAC can be implemented in the SP architecture in which the MAC and network protocol use the SP layer to perform their interactions. Figure 16 demonstrates the SP architecture containing IAMAC in itsMAC layer. By integrating IAMAC and SP we can apply cross-layer optimization while we also



(a)

(b)

Figure 15. Mean queue length per node with IAMAC as the MAC protocol. (a): Mean queue length for short Time/Super Frame durations (15 seconds and less). (b): Mean queue lengthfor long Super Frame durations (50 seconds and more).

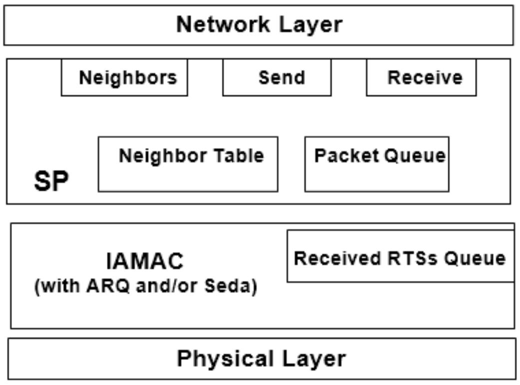


Figure 16. Implementing IAMAC within SP architecture. Network layer protocol and IAMAC can access to the Neighbor Table and Packet Queue data structures. Through threemain operations of SP (i.e., Neighbors, Send, and Receive), neighbor table can be managedand data packets can be sent or received via the MAC protocol. Maintain the modularity of the architecture. As a result, improvement of IAMAC or inclusion of other network protocols can be achieved easily in the future.

## 6. CONCLUSION

In this paper, we introduce a new medium access control protocol (IAMAC) designed to enhance the performance of wireless sensor networks in terms of both lifetime and delay. IAMAC achieves its high performance through three key mechanisms. First, it reduces inter-node interference and packet corruption with two interference avoidance algorithms. Second, it adopts a tree routing structure, allowing multiple nodes to transmit to a common parent during a Time/Super Frame, which minimizes control packet overhead and reduces per-hop latency. Third, IAMAC is a sleep/wake MAC protocol that decouples Time/Super Frame duration from synchronization, enabling a trade-off between lifetime and delay based on application requirements. We performed extensive simulations with a realistic data link model to assess IAMAC's performance. The results show that IAMAC offers a longer lifetime than S-MAC and Adaptive S-MAC, while its end-to-end latency is lower than that of S-MAC. Therefore, IAMAC is a suitable choice for applications with critical lifetime requirements, such as surveillance and monitoring. Additionally, IAMAC’s Time and Super Frame structures provide greater flexibility than S-MAC and Adaptive S-MAC. Finally, we demonstrated that by integrating IAMAC into the SP architecture, it can facilitate inter-layer interactions through the SP abstract layerThe simulations revealed that several parameters affect IAMAC’s performance, such as the duration of contention slots (RTS Slot and CTS Slot) and transmission power, which significantly influence network lifetime and latency. While simulations can help identify optimal values for these parameters, the process is complex and time-consuming. Thus, developing an analytical method for determining these optimal values could be beneficial. Input parameters for the analytical model include node density, sampling rate, and certain physical layer characteristics.

## REFERENCES

[1] M. Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," ACM Trans. Sen. Netw., vol. 3, 2007, p. 7.

[2] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," Proceedings of the 1st international conference on Embedded networked sensor systems, Los Angeles, California, USA: ACM, 2003, pp. 14-27.

[3] Q. Cao, T. He, L. Fang, T. Abdelzaher, J. Stankovic, and S. Son, "Efficiency centric communication model for wireless sensor networks," Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, IEEE, 2006, pp. 1-12.

[4] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," Proceedings of the 4th international symposium on information processing in sensor networks, IEEE Press, 2005, p. 11.

[5] D. S. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Wirel. Netw., vol. 11, 2005, pp. 419-434.

[6] J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks,"

[7] IEEE INFOCOM, New York, USA: 2002, pp. 1567-1576.

[8] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," Proceedings of the first international conference on Embedded networked sensor systems SenSys '03, New York, New York, USA: ACM Press, 2003, p. 171.

[9] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACM Trans. Netw., vol. 12, 2004, pp. 493-506.

[10] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04, New York, New York, USA: ACM Press, 2004, p. 95.

[11] M. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," IEEE INFOCOM 2004, IEEE, 2004, pp. 1740-1750.

[12] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," Wireless Communications, IEEE, vol. 12, 2005, pp. 3-11.

[13] V. Srivastava and M. Motani, "Cross-layer design: A survey and the road ahead," Communications Magazine, IEEE, vol. 43, 2005, p. 112,119.

[14] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05, New York, USA: ACM Press, 2005, p. 76.

[15] N. Ramanathan, M. Yarvis, J. Chhabra, N. Kushalnagar, L. Krishnamurthy, and D. Estrin, "A stream-oriented power management protocol for low duty cycle sensor network applications," Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors, IEEE Computer Society, 2005, pp. 53-61.

[16] J. van Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks,"

[17] Proceedings of the 2nd ACM international conference on wireless sensor networks and applications WSNA '03, New York, USA: ACM Press, 2003, p. 11.