

## RR-MAC: Receiver Reservation MAC Protocol with Time-Division for Wireless Sensor Networks

M. K. Kirubakaran<sup>1</sup>, Dr. N. Shankar Ram<sup>2</sup>

<sup>1</sup> Sathyabama University College, Chennai, India Head, R.M.D Engineering Professor & Chennai, India

---

**ABSTRACT:** *Wireless sensors networks are potentially employed in various fields such as defense system, target tracking, target monitoring, wildlife monitoring and disaster management. The node's of a wireless sensor network consists of three main parts- processor, transmitter/receiver and memory unit. For these parts to be operational in an efficient way, the battery power of the nodes must be prolonged and utilized effectively. Idle listening, control data overhead, collisions, retransmission are the major problems which reduce the energy efficiency in the nodes. This paper aims at proposing and analyzing RR-MAC protocol (Receiver-Reservation MAC protocol) which uses dynamic multichannel and time division algorithm to enable nodes utilize energy in efficient manner. Further, the idle time is considerably eliminated by adapting a central memory register for making reservations on receivers. The node's will be active only when needed and sleep rest of the time. Collision avoidance and energy loss due to retransmissions are addressed substantially in RR-MAC design. Since the collisions are reduced, the data can be sent with minimized time delay caused by retransmissions.*

**Keywords:** *Duty Cycling, Energy Efficiency, Interference, Medium access control, pseudorandom sequence*

---

### I. INTRODUCTION

Wireless Sensor Networks (WSNs) have emerged as one of the dominant technology trends of this decade (2000-2012) that has potential usage in defence and scientific applications. These WSNs can be used for different purposes such as target tracking, intrusion detection, wildlife habitat monitoring, climate control and disaster management [1]. A typical node in the WSN consists of a sensor, embedded processor, moderate amount of memory and transmitter/receiver circuitry. These sensor nodes are normally battery powered and they coordinate among them selves to perform a common task. These Wireless Sensor Networks have severe resource constrains and energy conservation is very essential. The sensor node's radio in the WSNs consumes a significant amount of energy. Substantial research has been done on the design of low power electronic devices in order to reduce energy consumption of these sensor nodes. Because of hardware limitations further energy efficiency can be achieved through the design of energy efficient communication protocols. Medium access control (MAC) is an important technique that ensures the successful operation of the network. One of the main functions of the MAC protocol is to avoid collisions from interfering nodes. The classical IEEE 802.11 MAC protocol for wireless local area network wastes a lot of energy because of idle listening. Designing power efficient MAC protocol is one of the ways to prolong the life time of the network. In this work the study of the energy efficient MAC protocols for the wireless sensor network.

### II. Mac Protocol Design Challenges

The medium access control protocols for the wireless sensor network have to achieve two objectives. The first objective is the creation of the sensor network infrastructure. A large number of sensor nodes are deployed and the MAC scheme must establish the communication link between the sensor nodes. The second objective is to share the communication medium fairly and efficiently.

#### 2.1 Attributes of a Good MAC Protocol

To design a good MAC protocol for the wireless sensor networks, the following attributes are to be considered [2]. (i) Energy Efficiency: The first is the energy efficiency. The sensor nodes are battery powered and it is often very difficult to change or recharge batteries for these sensor nodes. Sometimes it is beneficial to replace the sensor node rather than recharging them.

(ii) Latency: The second is latency. Latency requirement basically depends on the application. In the sensor network applications, the detected events must be reported to the sink node in real time so that the appropriate action could be taken immediately.

(iii) Throughput: Throughput requirement also varies with different applications. Some of the sensor network application requires to sample the information with fine temporal resolution. In such sensor applications it is better that sink node receives more data.

(iv) Fairness: In many sensor network applications when bandwidth is limited, it is necessary to ensure that the sink node receives information from all sensor nodes fairly. However among all of the above aspects the energy

efficiency and throughput are the major aspects. Energy efficiency can be increased by minimizing the energy wastage.

2.2 Major Sources of Energy Wastes Major sources is overhearing, meaning that a node picks up packets that are destined to other nodes.

(iii) Packet Overhead: The third source is control packet overhead. Sending and receiving control packets consumes energy too and less useful data packets can be transmitted.

(iv) Idle listening: The last major source of inefficiency is idle listening, listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, the sensor node will be in idle state for most of the time. The main goal of any MAC protocol for sensor network is to minimize the energy waste due to idle listening, overhearing and collision.

### 2.3 MAC Performance Matrices

In order to evaluate and compare the performance of energy conscious MAC protocols, the following matrices are being used by the research community.

(i) Energy Consumption per bit: - The energy efficiency of the sensor nodes can be defined as the total energy consumed / total bits transmitted. The unit of energy efficiency is joules/bit. The lesser the number, the better is the efficiency of a protocol in transmitting the information of energy waste in wireless sensor network are basically of four types [2] [3].

(i) Collision: The first one is the collision. When a transmitted packet is corrupted due to interference, it has to be discarded and the follow on retransmissions increase energy consumption. Collision increases latency also.

(ii) Overhearing: The second in the network. This performance matrices gets affected by all the major sources of energy waste in wireless sensor network such as idle listening, collisions, control packet overhead and overhearing.

(ii) Average Delivery Ratio: - The average packet delivery ratio is the number of packets received to the number of packets sent averaged over all the nodes.

(iii) Average Packet Latency: - The average packet latency is the average time taken by the packets to reach to the sink node.

(iv) Network Throughput:-The network throughput is defined as the total number of packets delivered at the sink node per time unit.

Wireless sensor network MAC protocols must operate under a number of significant challenges due to the shared nature of the wireless channel. For example, hidden terminals may cause wireless collisions, and heavy wireless usage in one location may create substantial contention for channel access by other nearby sensor nodes, degrading performance for these sensor nodes as well. Many other sources of wireless interference are also possible, including external interference from wireless transmissions by other types of devices, such as WI-Fi nodes [4]. In some cases, wireless interference may even be deliberate, such as from a malicious node performing an active jamming attack, transmitting continuously in order to block other nodes access to the channel [5].

## III. Related Work

Many energy-efficient sensor network MAC protocols have been proposed that use only a single radio channel for all transmissions; examples include S-MAC, B-MAC[13], X-MAC, DW-MAC, RI-MAC, and PW-MAC. The network throughput with these protocols is limited, however, to the capacity of a single channel, and their performance may be further substantially reduced under conditions of wireless interference or jamming.

Compared with these protocols, EM-MAC is not only more robust against such conditions but is also more capable of handling large and dynamic traffic loads by efficiently utilizing multiple channels. The opportunities possible by utilizing multiple orthogonal radio channels in the context of MAC protocols for wireless *ad hoc networks* using WI-Fi radios have long been recognized; examples include SSCH and McMAC. As these protocols were designed for a network in which nodes are always-on, however, energy efficiency was not a major concern in their design. In the context of *sensor networks*, researchers have recently also been exploring techniques at the MAC layer to utilize multiple orthogonal radio channels, such as are available in the IEEE 802.15.4 (ZigBee) transceivers widely used in current sensor networks. Example protocols in this area include Y-MAC, A-MAC, MMAC and CAM-MAC. In these protocols, prior to each data packet transmission, the sender and receiver node first tune to a dedicated *control channel* to negotiate the channel to use for the data transmission. Negotiating first on a control channel makes it easier for a sender to rendezvous with a receiver for transmitting a data packet, but since no data packet transmission would then be possible without a successful negotiation on the control channel, the available bandwidth of the control channel can become a packet transmission bottleneck. Furthermore, if the control channel is subject to heavy interference such as from ZigBee or WI-Fi traffic [6] or is jammed by an attacker [7], such protocols would be unable to deliver any data packets. In addition, the energy efficiency of these protocols can deteriorate substantially from waiting for the

opportunity to negotiate a data channel using the control channel. In contrast, EM-MAC does not rely on a dedicated control channel and is adaptive to dynamic channel conditions, contributing to the high efficiency of EM-MAC and its resilience to wireless interference and jamming.

Like EM-MAC, the MuChMAC protocol [12] does not use a dedicated control channel; each node in MuChMAC independently decides its channel-switching sequences, which can be deduced by its neighbors. However, unlike EM-MAC, nodes in MuChMAC use a fixed set of channels, without adapting to current channel conditions such as interference or jamming on a channel. In addition, MuChMAC divides time into slots, assuming loose global time synchronization and a fixed upper bound on clock drift in attempting to ensure that sender and receiver at least both wake up within the same slot as each other; MuChMAC does not provide any mechanism to recover in the case in which the sender and receiver fail to rendezvous in this way due to larger clock drifts or variable hardware or operating system latencies. In contrast, EM-MAC operates entirely asynchronously, with no reliance on global time synchronization, and provides an efficient mechanism to quickly rendezvous a sender and a receiver after any rendezvous failure occurs. Some protocols have attempted to make use of multiple channels by assigning different channels to different nodes in a two hop neighborhood to avoid potential interference; examples of such protocols include MMSN [10] and TMMAC [19], in which time slots are used to coordinate transmissions. To enable such time slots, these protocols assume precise time synchronization in the network, which is not needed in EM-MAC. Similarly, TMCP [20] partitions a network into different trees and assigns a fixed channel to each tree. This design showed high performance in data collection applications, but the fixed channel assignment to each tree is inefficient in handling dynamic traffic and makes the traffic on a tree vulnerable to wireless interference and jamming attacks as discussed above. Finally, several other energy-efficient MAC protocols have used various forms of predictive wake-up based on pseudo random number sequences, similar to that used in EM-MAC [17]. For example, Cao et al. [14] presented an analytical study of different energy-efficient MAC protocol schemes based on globally synchronized time slots: time is divided into a sequence of frames and each frame is divided into a sequence of these slots. In their 'pseudo-random Staggered Onscheme, each node wakes up as a receiver independently in each slot with probability  $\psi r$  (eg.,  $\psi r = 0.01$ ), where the decision to wake up or not for each slot is determined by comparing the next number in that node's pseudorandom number sequence to this  $\psi r$  threshold. Through the use of the pseudorandom number sequence, the slot in which any node will be awake to receive can be predicted by a sender that knows and follows the state of that node's pseudorandom number generator. Cao et al. also provided a rough sketch of the operation of a protocol called O-MAC based on this scheme, although in O-MAC, rather than waking up with an independent probability in each slot, the wake-up time of a receiver is simply generated as the slot number within a frame, based on the next number in the pseudorandom sequence. The MAC protocol of the JAVeLENSystem [5] uses predictable receiver wake-up timing similar to the 'pseudo-random Staggered Onscheme of Cao et al. [14]; JAVeLEN divides time into globally synchronized slots, and each node independently wakes up to attempt to receive in each slot depending on the comparison of the next number in that node's pseudorandom sequence against a threshold.

### 3.1 EM-MAC Protocol

EM-MAC is a multichannel asynchronous duty-cycling MAC protocol. It does not require nodes to synchronize their clocks, does not use a common control channel, and does not explicitly exchange channel and wake-up schedules. Instead, every node independently decides its own *pseudorandom* channel-switching behavior and wake-up times. A sender rendezvous with a receiver by predicting the receiver's wake-up channel and wake-up time based on the sender's knowledge of the state of the receiver's pseudorandom function used to generate its wake-up channels and times. EM-MAC is a *receiver-initiated* MAC protocol; a node sends a wake-up beacon to notify potential senders that it is awake and ready to receive data packets. After receiving a wake-up beacon from a receiver  $R$ , a node  $S$  that has a data packet for  $R$  sends it to  $R$ .  $R$  sends an ACK beacon to  $S$  to acknowledge the data packet receipt and to allow another data packet to be sent to  $R$  by this or another sender; in this example, no other data packet is available, so  $S$  and  $R$  quickly go back to sleep. After  $R$  wakes up for the second time shown, no node has a data packet waiting to send to  $R$  and  $R$  quickly goes back to sleep. Finally, after  $R$  wakes up again,  $S$  has another packet for  $R$  and sends it to  $R$  in response to this beacon from  $R$ .

In order to reduce wireless collisions caused by nodes waking up at the same time and on the same channel and to distribute the traffic among the available channels, a node in EM-MAC switches among the channels it selects based on its pseudorandom channel scheduling. In addition, each node in EM-MAC pseudorandomly decides its own wake-up times; a wake-up time of a node is determined from the node's previous wake-up time plus its current pseudorandomly chosen wake-up interval. In particular, for each wake-up, a node invokes its pseudorandom number generator twice: once to compute its next wake-up channel and again to compute its next wake-up interval. In our experiments, node pseudorandom wake-up intervals range between 500 ms and 1500 ms. In a typical EM-MAC protocol [17], a Sender has previously learned the time and pseudorandom number generator information of  $R$  and is able to predict the wake up channels and wake-up times of  $R$ . When  $S$  has a data packet to send to  $R$ ,  $S$  wakes up on the predicted wake-up channel of  $R$  right before the predicted wake-up time of  $R$ , thereby achieving high energy efficiency by minimizing the idle

listening and overhearing. To avoid using congested channels and to be robust against wireless interference and jamming, a node dynamically modifies the set of channels among which it switches, based on the channel conditions it senses.

For energy-efficient resolution of wireless collisions, EM-MAC deploys the collision resolution mechanism, where in upon occurrence of a collision, a receiver notifies potential senders to retransmit their packets using an increased back off window via a new beacon once a collision is detected. If the collision resolution mechanism does not resolve the collision before the receiver goes to sleep, a sender in EM-MAC goes to sleep and wakes up again to retransmit at the receiver's next predicted wake-up time, if a predefined retry limit has not been reached. EM-MAC uses any pseudorandom function to generate the channel and wake-up schedule for a node.

### 3.1.1 Clear Channel Assessment

A node in EM-MAC independently collects the channel condition information, as a byproduct of regular transceiving operations, without extra energy consumption. EM-MAC does not send probing packets to determine the channel condition because such proactive channel condition measurement approach increases the node energy consumption and network traffic. A node maintains for each channel a non-negative "badness metric [17]. When a node in EM-MAC wakes up on a channel to send a wake-up beacon or a data packet, it conducts a CCA (Clear Channel Assessment) check to ensure the channel is idle before beginning the transmission. If the channel is idle, the node sends the packet and decreases that channel's badness metric by 1 (the metric is not allowed to become less than 0). Otherwise, the node conducts a CCA check again after a short random backoff. If the channel is still busy after three such CCA checks, the node increases that channel's badness metric by 2 and goes to sleep. In addition, after a node sends a wake-up beacon, if the CCA check indicates the channel is busy but the node does not receive a valid packet, the node assumes a packet collision may have occurred and resolves the collision by informing the senders to retransmit the packets using an increased backoff window. If collision resolution fails, the node increases its badness metric for this channel by 2. Likewise, if the node sends a data packet but does not receive an ACK for it, the node increases the badness metric of this channel by 2. By updating the channel badness metric in this way, if a channel is congested or many failed transmissions occurred on it, the badness metric of this channel increases; Otherwise, the channel's badness metric decreases. Based on the channel badness metrics, a node in EM-MAC selects the set of channels it switches among. Every node maintains its own *channel blacklist* that identifies the channels the node regards as bad channels. When the badness metric of a channel is above a threshold  $C_{bad}$ , the channel will be added to the node's blacklist. A node switches among channels based on its pseudorandom channel schedule, except that, if the next pseudorandomly chosen channel is on the node's channel blacklist, the node stays on its current channel (used for its wake-up previous to this one).

### 3.1.2 Exponential Chase Algorithm

If a sender and receiver have not communicated for a long time due to collision or when the sender misses the a wake-up of the receiver, as even a very small clock rate difference multiplied over a long time can result in a large prediction error, quickly re-rendezvousing with this receiver is crucial to maintaining a small sender duty cycle and packet delivery latency. EM-MAC introduces the *exponential chase* algorithm, a back-off mechanism to quickly re-rendezvous a sender and a receiver. After a sender misses a receiver for the second time, the sender invokes the exponential chase algorithm by doubling its current wake-up advance time for this receiver (and thus also the time to wait for the receiver's wake-up beacon after the expected receiver wake-up time). Then the sender computes the receiver's future wake-up channels and wake-up times until finding a wakeup time of the receiver that is at least this wake-up advance time after the receiver's current time. Lastly, the sender wakes up in the wake-up advance time before this predicted receiver wake-up time on the corresponding wake-up channel of the receiver, to attempt to rendezvous with the receiver.

### 3.1.3 Back-off iterations

The sender repeats iterations of this exponential chase algorithm until receiving a wake-up beacon from the receiver. Once the sender successfully rendezvous with this receiver, it resets its wake-up advance time for this receiver to its initial value for future data packets to be sent to this same receiver.

The above figure shows an example of the operation of exponential chase of EM MAC. The exponential chase algorithm ensures that a sender is able to rendezvous with a receiver after a finite number of iterations of the algorithm, as the time difference between the sender and the receiver must be finite. Meanwhile, a sender  $S$  gives up attempting to send to a receiver  $R$  after definite number of attempts and discards the prediction state of  $R$  if  $S$  has not been able to rendezvous with  $R$  in the exponential chase iteration. The reason to limit the sender in this way in the exponential chase algorithm is because the receiver node may have been powered off or be out of wireless transmission range. If, instead, the sender is able to rendezvous with the receiver, after receiving the receiver wake-up beacon, the sender sets a flag in the header of the data packet that

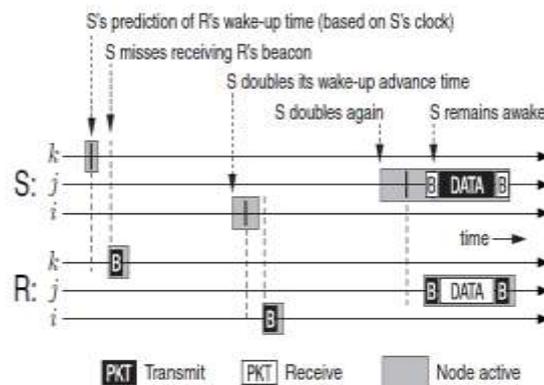
it sends to this receiver, requesting the receiver to send back its current prediction state in the ACK beacon; the sender thus regains its ability to precisely predict the wake-up times and wake-up channels of this receiver once receiving this prediction state.

#### IV. Rr -Mac Protocol Design

In RR-MAC(Receiver reservation MAC), there are multiple channels. Each node in the RRMAC uses a pseudorandom sequence to determine its wake-up channel(channel in which it becomes active) and another pseudorandom sequence to determine its wake-up time (time at which it has to be come active in the wake-up channel). The nodes of RR-MAC can look into the pseudorandom sequences of any other node. In other words, the nodes are initially fed about the other nodes in the network and what pseudorandom sequence is pursued in each node to obtain the wake-up channel and time. Thus a sender S which has to transmit data to a receiver R, will actually imitate the receivers pseudo function to know where the receiver is active in terms of channel and time. Then the sender will also wake-up in the same channel at the same time as the receiver. When no data is there to transmit, then the node simply goes with its own pseudorandom sequence to wake-up in the channel and time obtained with its pseudo sequence, expecting data from prospective senders, if any.

Fig 1: Sender S sends data packets to receiver R using

EMMAC. Only three of the channels are shown here, labeled *i*, *j*, and *k*. At the time of R's second beacon, no node has a packet waiting to send to R.



When there are two senders S1 and S2 to transmit data to receiving node R, then both senders imitate the pseudo sequence at the receiver node to obtain the receiver's next wake-up channel and next wake-up time. Since, there is no control channel here, both senders are unaware of each other. They may both imitate the receiver and wakeup simultaneously at the receiver's channel time (little earlier than actual wake-up time so as not to miss the receiver's "Ready" beacon). When the receiver wakes up in the channel, there are also two senders waiting out there to reach the receiver. The receiver follows the standards of transmission, wherein, it sends a "Ready" beacon first to indicate its availability to any prospective senders. The two senders upon getting this "Ready" beacon from the receiver, come to know the active existence of receiver and begin their transmission. This leads to collision of packets from the two senders. In EM-MAC the collision is attempted to be resolved by exponential back-off algorithm. Upon a collision, the senders will back-off and retransmit the packets after their respective back-off time. The back-off time is calculated individually at each sender. Thus the two senders are likely to retransmit after different back-off time intervals and collision is avoided. This technique serves well under low traffic conditions. Under high traffic there can be large number of senders to one receiver. Upon collision, all of them back-off and reattempt transmission. There is a high probability that after first back-off, still some nodes undergo collision or sense the channel busy and again back-off second time. Gradually, as the back-off continues, slowly each sender will be able to access channel without collision and close the transmission one by one. As it can be seen, the senders undergoing collision have to attempt back-off and retransmit. Under high traffic, a sender may have to back-off several times, before succeeding in its transmission. This also adds time delay along with energy drain for iterative retransmissions.

To address these two problems - energy waste and time delay due to back-offs, RR algorithm is devised. In RR (Receiver Reservation) algorithm each sender which has data to transmit to a receiver, after imitating the wakeup channel and wakeup time of the receiver, will have to make reservation on the receiver. Sender *s1* after knowing the receiver's wakeup channel and wake up time  $t_r$  reserves the receiver for a time  $t_s$  (depending on the data to send), after receiver wakeup time  $t_r$ . Thus the receiver when it wakeup at time  $t_r$  will receive the packets from sender S1 from time  $t_r$  till  $(t_r+t_s1)$ . Similarly, the second sender *s2* will reserve the receiver for a time of  $t_s2$ . Seeing that the receiver is reserved already from  $t_r$  till  $(t_r+t_s1)$ , it will have to make its

reservation from  $(t_r+t_{s1})$  till  $(t_r+t_{s1}+t_{s2})$ . A common central memory registry is maintained wherein the senders make reservation for a receiver.

When the receiver wakes up at time  $t_r$ , the senders wakeup one by one according to the reservation they made in the registry, thus bringing out a discipline among the senders to transmit to the receiver. This RR algorithm potentially resolves the energy waste and time delay problems explained earlier due to back-off reattempts.

The receiver upon seeing the reservation made in the control registry will already be aware how long it will have to be listening after its wake up time  $t_r$ . This removes the burden of the receiver to ensure no other sender waits out there to reach it, before going to sleep. Thus the energy is saved at the receiver end also.

#### 4.1 RR-Algorithm

1. Sender S1 contains data to be transmitted to receiver R.
2. S1 imitates the pseudo sequence of R to know R's wake up channel and wake up time  $t_r$ .
3. S1 makes reservation of the receiver R for a time starting from  $t_r$  till  $(t_r+t_{s1})$ ,  $t_{s1}$  depends on amount of data S1 has to transmit to R.
4. This reservation details are entered in the central registry.
5. Another sender S2 to transmit to R imitates the wake up channel and time. Then, it enters its reservation from time  $(t_r+t_{s1})$  till  $(t_r+t_{s1}+t_{s2})$ .
6. The next sender, if any, make reservation in the registry for its transmission respectively.

RR MAC is implemented with a typical network comprising of two sources transmitting to a receiver. The transmission speed of the network is held at 100 kbps, for ease of calculation. The sleep time is subjected to a variation of 1000 ms to 3000 ms. The proposed protocol and existing protocol are applied onto the network to obtain the number of packets transmitted in course of time.

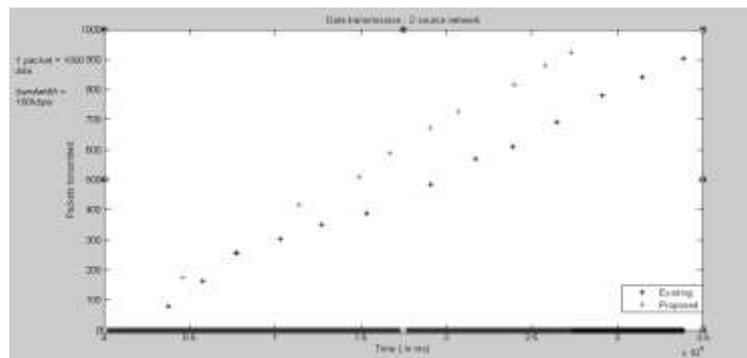


Fig 2: Two sources transmitting packets to receiver.

As can be seen in fig 2 RR-MAC enables the nodes to transmit more packets in the same time as compared to EM-MAC protocol.

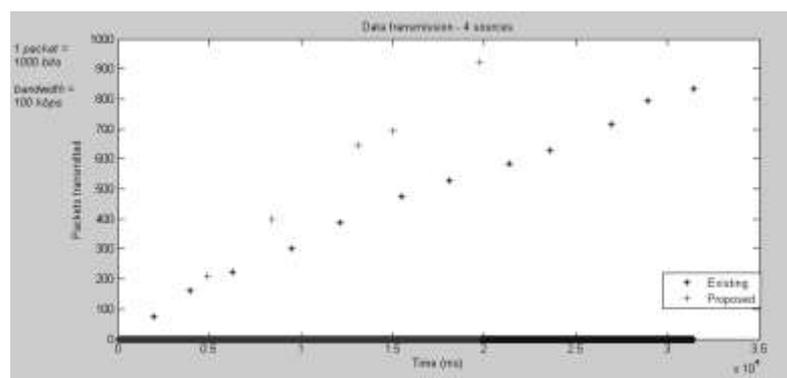


Fig 3: Four sources transmitting packets to a receiver

As the number of nodes is increased to four, the RR MAC gives still better results. This can be accounted by the fact that, when collision occurs all four sources back-off and reattempt to transmit independently. It takes course of time for all four nodes to pick up the channel when it is free and complete transmission to the receiver. In RR MAC, the four nodes complete their transmission without any back-off, each completing the transmission as per the reservation made on the receiver. The packets transmitted in the four source network is depicted in figure 3.

#### IV. Conclusion

By adapting time division on the receiver wake up time, sender nodes make a regulated transmission to the receiver. It is a win-win situation for both the senders and receiver. In future work, this paper will extend the usage of memory register to proactively decide if a receiver has to wake up in its next wake up time or can skip the next wake up when no reservation is made by any sender to transmit in the next wake up time. This will save energy at the receiver node by eliminating unnecessary wake ups.

A bursting algorithm is also to be devised so that a sender can transmit data partially to the receiver and transmit the remaining packets to the receiver in the next wakeup. This algorithm will be adopted especially when there is high data traffic and the data size is huge at the sender. Then, in such scenario, one sender may reserve the receiver for a very long time to complete its huge data transmission. This will cause the receiver potentially unavailable to other senders for a long duration. To avoid this, the sender can split the huge data into multiple transmissions. The huge data is split and transmitted across successive wake ups of the receiver. This ensures that no single sender will reserve a receiver for a very long time causing serious delay for other awaiting senders.

#### V. Acknowledgments

We thank software analyst Sasidharan, for guiding us through the paper development. We also thank the research colleagues for their feedback on the content.

#### References

- [1] Rajesh Yadav, Shirshu Varma and N.Malaviya:Optimized Medium Access Control for Wireless Sensor Network,IJCSNS International Journal of Computer Science and Network Security, Vol. 8, No.2, pp. 334 -338 (February 2008).
- [2] V. Rajendran, K. Obraczka and J.J. Gracia-Luna-Aceves: Energy Efficient, Collision Free Medium Access Control for Wireless Sensor Networks, in ACM International Conference on Embedded Networked Sensor Systems (SenSys), pp. 181-192 (November 2003). [3] M.Ali, Saif, A. Dunkels, T. Voigt, K. Romer,K. Langendoen, J. Polastre, Z. A. Uzmi:Medium Access Control Issues in Sensor Networks, ACM SIGCOMM Computer Communication Review, Vol. 36, No. 2 (April 2006).
- [4] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the Second ACM Conference on Embedded Network Sensor Systems (SenSys 2004)*, pages 95–107, November 2004.
- [5] J. Redi, S. Kolek, K. Manning, C. Partridge, R. Rosales-Hain,R. Ramanathan, and I. Castineyra. JAVeLEN—An Ultra-Low Energy Ad Hoc Wireless Network. *Ad Hoc Networks*, 6:108–126, January 2008.
- [6] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *Proceedings of the 8th ACM Conference on Embedded Network Sensor Systems (SenSys 2010)*, pages 309–322, November 2010.
- [7] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, pages 46–57, May 2005.
- [8] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *Proceedings of the 8th ACM Conference on Embedded Network Sensor Systems (SenSys 2010)*, pages 309–322, November 2010.
- [9] Tie Luo, Mehul Motani, and Vikram Srinivasan. Cooperative Asynchronous Multichannel MAC: Design, Analysis, and Implementation. *IEEE Transactions on Mobile Computing*, 8(3):338–352, March 2009.
- [10] Gang Zhou, Chengdu Huang, Ting Yan, Tian He, John A. Stankovic, and Tarek F. Abdelzaher. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In *Proceedings of the 25<sup>th</sup> IEEE International Conference on Computer Communications (INFOCOM 2006)*, pages 1–13, April 2006.
- [11] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: Slotted Seeded Channel Hoppingc for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *Proceedings of the 10<sup>th</sup> Annual International Conference on Mobile Computing and Networking (MobiCom 2004)*, pages 216–230, October 2004.
- [12] Joris Borms, Kris Steenhaut, and Bart Lemmens. Low-Overhead Dynamic Multi-channel MAC for Wireless Sensor Networks. In *Proceedings of 7th European Conference on Wireless Sensor Networks (EWSN 2010)*, pages 81–96, February 2010.
- [13] 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 ACM Conference on Embedded Network Sensor Systems (SenSys 2006)*, pages 307–320, November 2006.
- [14] Hui Cao, Kenneth W. Parker, and Anish Arora. O-MAC: A Receiver Centric Power Management Protocol. In *Proceedings of the 14<sup>th</sup> IEEE International Conference on Network Protocols (ICNP 2006)*, pages 311–320, November 2006.
- [15] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan (Mike) Liang, and Andreas Terzis. Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless. In *Proceedings of the 8th ACM Conference on Embedded Network Sensor Systems (SenSys 2010)*, pages 1–14, November 2010.
- [16] Amre El-Hoiydi and Jean-Dominique 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 2004)*, pages 18–31, July 2004.
- [17] Lei Tang, Yanjun Sun, Omer Gurewitz, David B. Johnson. EMMAC: A Dynamic Multichannel Energy-Efficient MAC Protocol for Wireless Sensor Networks, in ACM SIGMOBILE International Conference on Mobile Ad Hoc Networking and Computing, pp. 181-192 (November 2003).
- [18] Youngmin Kim, Hyojeong Shin, and Hojung Cha. Y-MAC: An Energy-Efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks. In *Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 53–63, April 2008.
- [19] Jingbin Zhang, Gang Zhou, Chengdu Huang, Sang H. Son, and John A. Stankovic. TMMAC: An Energy Efficient Multi-Channel MAC Protocol for Ad Hoc Networks. In *Proceedings of the 2007 IEEE International Conference on Communications (ICC 2007)*, pages 3554–3561, June 2007.
- [20] Yafeng Wu, John A. Stankovic, Tian He, Jiakang Lu, and ShanLin.Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM 2008)*, pages 1193–1201, April 2008.