ETEEM- Extended Traffic Aware Energy Efficient MAC Scheme for WSNs

Younas Khan[1]
Department of Computer Sciences
Institute of Management Sciences
Peshawar, Pakistan

Sheeraz Ahmed[3]
Department of Electrical Engg
University of Engg & Technology
Peshawar, Kohat, Pakistan

Saqib Shahid Rahim[5]
Department of Computer Sciences
Abasyn University
Peshawar, Pakistan

Fakhri Alam Khan[2]
Department of Computer Science
Institute of Management Sciences
Peshawar, Pakistan

Imran Ahmad[4]
Department of Computer Sciences
Institute of Management Sciences
Peshawar, Pakistan

M. Irfan Khattak[6]
Department of Electrical Engg
University of Engg & Technology
Peshawar, Kohat, Pakistan

***Abstract*—Idle listening issue arises when a sensor node listens to medium despite the absence of data which results in consumption of energy. ETEEM is a variant of Traffic Aware Energy Efficient MAC protocol (TEEM) which focuses on energy optimization due to reduced idle listening time and much lesser overhead on energy sources. It uses a novel scheme for using idle listening time of sensor nodes. The nodes are only active for small amount of time and most of the time, will be in sleep mode when no data is available. ETEEM reduces energy at byte level and uses a smaller byte packet called FLAG replacing longer byte SYNC packets of S-MAC and SYNCrts of TEEM respectively. It also uses a single acknowledgement packet per data set hence reducing energy while reducing frequency of the acknowledgment frames sent. The performance of ETEEM is 70% better comparative to other under-consideration MAC protocols.**

Keywords—Energy Consumption; Multi-hop; Network Allocator Vector; Throughput; Wireless Sensor Networks

I. Introduction

Wireless sensor networks (WSNs) [1] are used in scenarios where human body cannot reach easily like earthquakes, battlefield, flood forecasting or for weather forecasting etc. WSN is a wireless network consisting of autonomous devices or nodes using sensors to monitor physical or environmental changes [2]. Each wireless node consists of sensor [3], radio transceiver [4], microcontroller [5], buffer [6] and power source [7]. Sensor is used to sense data in proximity. Transceiver is used to receive and send data to other sensor nodes. When data is sensed by sensor, then node becomes active (i.e. its state changes from sleep state to active state), processes that data and forwards it as specified by protocol. WSN uses adhoc topology [8] and most of the time the nodes are scattered on geographical area. Each node uses its own power source to operate. Wireless sensor nodes may use multi hop communication [9], in which one sensor node may pass data to another which in turn may forward it to other nodes till it reach to base station. Power consumption of wireless nodes depends on node operations. Power is critical to communication [10]. Four major sources of power wastage [10] in WSN are idle listening time, Overhearing, Overhead and Hidden terminal problem.

Idle listening problem arises when a sensor node listens to medium despite the absence of data which results in consumption of energy. Overhearing occurs in multi hop communication in which a node senses data which is destined to another node. Similarly, overhead occurs as a result of overhearing.

Several techniques, such as S-MAC [10], T-MAC [11], RMAC [12] TEEM [13] etc have been used to optimize energy consumption of WSN node. Sensor-MAC (S-MAC) was proposed to minimize energy consumption for all four sources (i.e. idle listening time, overhearing, overhead, hidden terminal problem) of power wastage. One major drawback of S-MAC protocol is its fixed time interval of active and sleep state which may still result in idle listening time. To address this issue, Traffic Aware Energy Efficient MAC (TEEM) protocol was proposed. TEEM protocol uses variable time for active and sleep state depending upon traffic instead of fixed interval. TEEM protocol reduces energy from idle listening and overhead part of energy consumption sources. When there is data with the nodes then it is active for whole life cycle else it goes to sleep state very earlier then its normal schedule. Both S-MAC and TEEM protocols send Acknowledgment (ACK) packet per fragment. TEEM consumes 16% less energy than S-MAC; but still it wastes energy in SYNCdata part i.e. if there is no data then it still waits for SYNCnodata part. Further, ACK packet sent per fragment also consumes energy. Due to these issues in TEEM protocol and S-MAC protocols we developed Extended Traffic Aware Energy Efficient MAC (ETEEM) protocol. ETEEM protocol not only incorporates variable time slot for active and sleep interval, but also introduces ACK packet per burst instead of ACK packet per fragment; resulting in reduced overhead. It uses four- byte packet called FLAG instead of ten byte SYNC packet. Major contributions of this paper include:

1) Energy optimization due to reduced idle listening time
2) Energy optimization due to reduced overhead
3) Implementation of ETEEM protocol, simulation and evaluation

4) Comparison of ETEEM protocol with S-MAC protocol and TEEM protocol

II. LITRATURE REVIEW

Initially 802.11 wireless standards IEEE [14] were developed to deal with wireless networks. CSMA/CA [15] under 802.11 standards was considered as the standard protocol. CSMA/CA is used for mobility (cellular) in wireless networks. It uses RTS/CTS and ACK technique for contention for the medium. It cannot be used in WSN because this standard gives no service for power saving and scalability for multi hop communication.

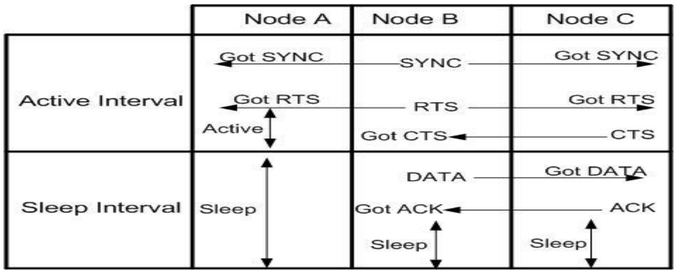

Fig. 1. S-MAC-When node has some data to send

In S-MAC protocol, sensor node first sends SYNC packet to form virtual clustering and CTS/RTS/ACK technique for contention for the medium, and subsequently makes communication with the right user. If a sender occupies the medium, next step is to send RTS packet, RTS packet has the corresponding receiver node address with which sender wants to communicate. When RTS reaches to receiver, it finds its address inside the packet and recognizes that sender node wants to communicate with him. As a result receiver node sends CTS packet. CTS mean that receiver node is ready to catch sender packets. Figure 1[13] shows S-MAC protocol behavior when there is data with one node and it want to send it to other node. Nodes will remain active in sleep state if data arrives and will go to sleep state as soon as data transfer is completed. It behaves differently when node does not have any data transfer with other nodes as showed in figure 2 [13].

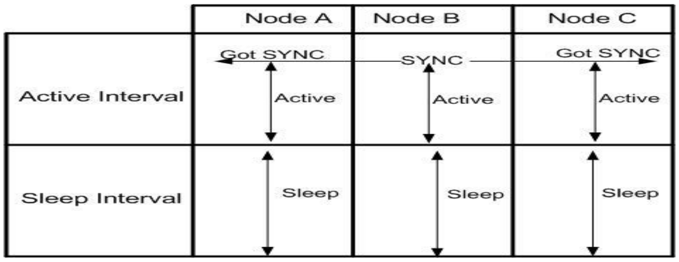

Fig. 2. S-MAC-When node has no data to send

S-MAC divides the whole life of each node into two parts: Listen and Sleep interval. In listen interval, wireless nodes can take part in communication with other nodes; and in sleep interval, it turns off its transceiver and does not listen to any data unless sleep time expires. Listen interval of sensor node is further divided into two parts: SYNC and DATA as in figure 2. If a node joins the network, it waits for random amount of time. However, if it does not listen to any schedule broadcasted by other nodes it can choose its own schedule and broadcast it in SYNC packet to its immediate neighbors, which further broadcasts it and eventually reaches to all nodes.

S-MAC uses fixed time intervals for listen and Sleep state. When node is in active state, it sends data in burst but if a node does not has any data to send then it must wait for whole listen interval to go to sleep state. Due to fixed time interval, it consumes energy because nodes wait for the whole listen time to over. T-MAC protocol, inspired by S-MAC protocol takes work further and makes communication between two nodes in WSN depends on time. T-MAC protocol does not uses fixed active and sleep interval. It allows nodes to stay awake until an activation event has occurred for certain amount of time. Apart from this, in T-MAC protocol nodes sends data in burst and go to sleep state between bursts which minimize idle listening time. It also uses FRTS technique. FRTS stands for Future Request To Send. T-MAC protocol increases overhead by including FRTS mechanism which consumes energy. If two or more nodes send FRTS frame at same time then there is possibility of collision which waste energy.

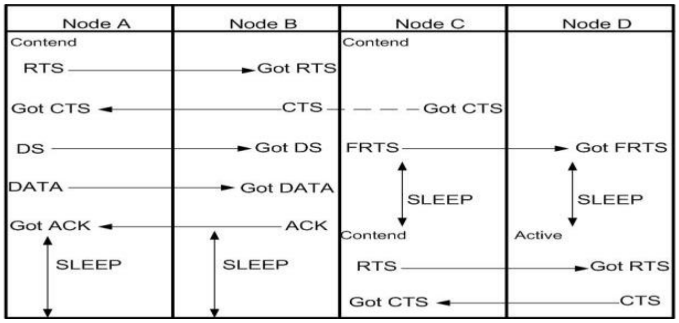

Fig. 3. FRTS frame of T-MAC

R-MAC protocol inspired by FRTS technique uses a special packet called POIN instead of RTS and CTS packet. Like S-MAC it divides nodes life cycle in three parts: SYNC, DATA and SLEEP. R-MAC protocol ensures that nodes will send data in sleep period. POIN is initiated by multi-hop which informs all nodes on path as when to wake up in sleep period. R-MAC protocol improves latency as compared to S-MAC and T-MAC protocol but it cannot guarantee collision avoidance. Two nodes when initiate POINS may succeed in transmitting POINS but both of them will start communication to their intended receiver at the start of sleep period and hence collision may occur. It wastes large percentage of energy in collision. TEEM [13] makes periodic awake and sleep intervals depending on traffic. The basic function is when there is traffic with node, then it sends it in burst in SYNCdata , and if there is no data to sends then it only send SYNC packet and goes to sleep state. Hence, it can minimize half the power compared to S-MAC. TEEM protocol divides sensor node lifecycle into two parts: SYNCdata and SYNCnodata. Nodes can send data in SYNCdata part and if does not have any data to send, then goes to sleep state in SYNCnodata part; too early then Sleep state as in figures 4 and 5. It also uses SYNCrts packet in contention for medium. RTS packet combined with SYNC is used to minimize overhead.

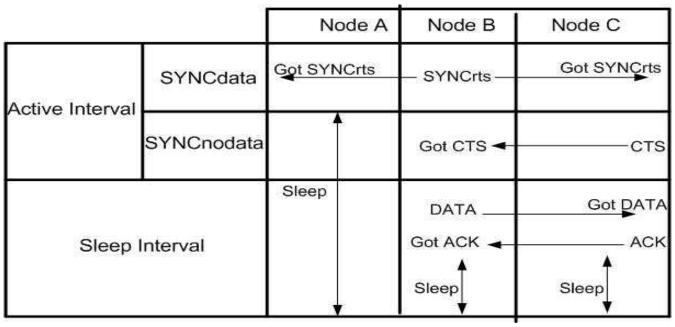

Fig. 4. TEEM-When node has some data to send

From previous figure, it is clear that TEEM protocol saves energy from idle listening time and overhead sources of energy consumption. But if we look closely, TEEM protocol also wastes energy in idle listening. If nodes do not have any data to send, then they waste first half that is SYNCdata part in idle listening. It is so because nodes will trigger their sleep schedule in SYNCnodata part.

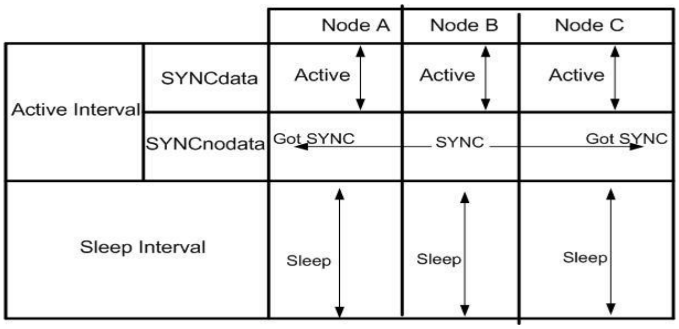

Fig. 5. TEEM-When node has no data to send

III. Proposed ETEEM Model

The proposed framework ensures that wastage of energy consumed by previous protocols in various ways is minimized. It saves energy from idle listening and overhead part of power consumption sources. ETEEM protocol neither has SYNCnodata, SYNCdata parts like TEEM protocol nor fixed listen and sleep time like S-MAC protocol. When node sleeping quantum expires and has no data to transfer or receive, then it waits for random amount of time and sends FLAG with value 0. If it has data to send, then it sends SYNCrts packet. In ETEEM protocol, nodes will not wait any more and will not listen to idle medium. S-MAC protocol sends separate SYNC and RTS packets which consume too much energy. It also sends ACKs per fragment which waste energy. TEEM protocol combines SYNC and RTS packet called as SYNCrts packet as shown in figure 5 [15]. Through this technique TEEM minimizes wastage of energy but TEEM protocol also supports ACKs per fragment technique which waste too much energy like S-MAC. ETEEM protocol, on one hand sends SYNCrts packet combining SYNC and RTS packet hence saves energy. On the other hand, it sends only one ACK for whole burst. There is no need to send ACKs per fragment. ETEEM protocol saves energy from this part also. The whole burst will be acknowledged in one ACKs packet.

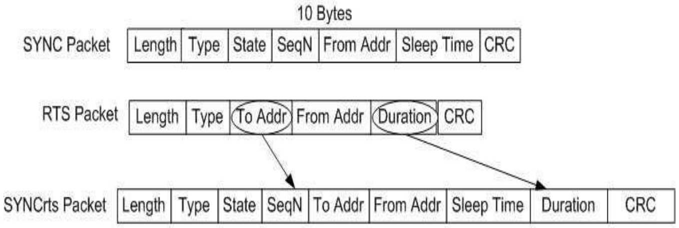

Fig. 6. SYNCrts packet

1) *SYNCrts*

WSN uses adhoc topology some nodes may die or disconnected while some may join later. Therefore, every wireless sensor network protocol should be capable to handle Synchronization. ETEEM protocol support Synchronization the same way as TEEM does i.e. through SYNC packet. In ETEEM protocol, when new node joins topology, it waits for some time and listens to the medium to see any of its neighbors broadcasted its schedule or not. If it hears schedule broadcasted by its neighbor, it simply adopts its schedule and stores it in schedule table. If there is no schedule broadcasted by its neighbor then it chooses its own listen and sleeps time and broadcast it to its immediate neighbors.

2) *Adding Extra Packet*

TEEM protocol divides Listen interval of nodes in two parts: SYNCdata and SYNCnodata. If node has data to send then it sends it in SYNCdata part, and if it does not have any data to send then it waits for SYNCnodata part. In SYNCnodata part of listen interval it sends SYNC packet to its immediate neighbor indicating that it has no data and go to sleep state. If we examine closely, TEEM protocol wastes first half i.e SYNCdata part. I f node does not have data to send, then it just consumes energy while sending no data to other nodes. ETEEM enhances this portion of TEEM protocol. ETEEM uses one extra packet, called FLAG having two fields: Data and Length as shows in Figure 6. Data has two possible values either 0 or 1. If Data has 0 value, it means node has no data, if it is 1 then the receiver must check the length field which will show the duration of the traffic.

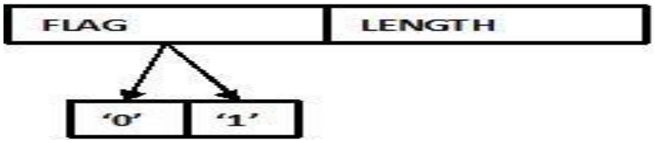

Fig. 7. FLAG packet with values 0 and 1

If node sleep time expires, it waits for random amount of time. After certain time, if it does not have any data to send, it will send FLAG packet with data value 0 and go to sleep state. If it has data to send, it will send RTS packet. After receiving CTS packet from the receiver, normal communication begins. After last packet, if it has further data arrived from other nodes and wants to send to the same receiver, then it will send FLAG with data value 1 indicating that it has more data to send. Receiver will resend RTS packet and further communication starts again. The receiver resends RTS packet for its neighbor nodes. It is possible that its neighbors sleep time expires and come to listen state, then through RTS packet they will come to know that medium is still busy. Therefore, they will again go to sleep state, otherwise, will send FLAG with Data value 0 indicating no further data and will go to sleep state early than its schedule, hence saving energy. It consumes less energy as compared to S-MAC and TEEM as depicted in figures 8 to 10.

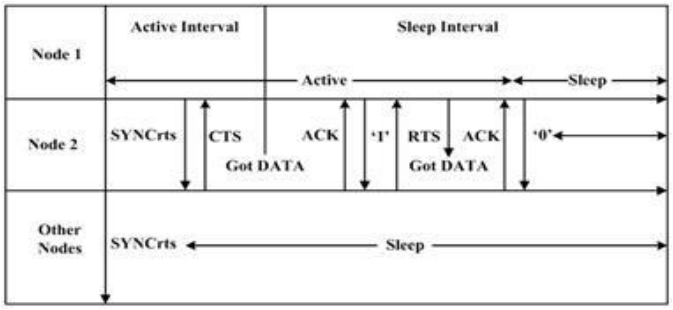

Fig. 8. when there is further data to send

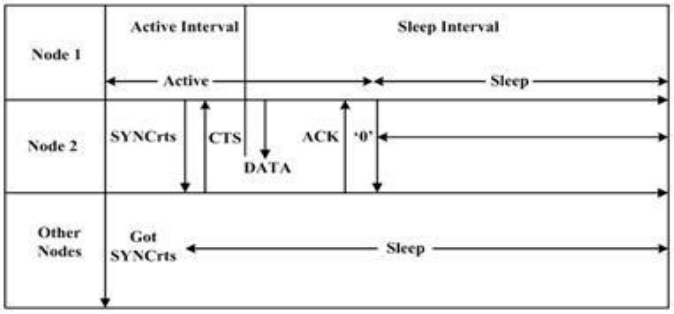

Fig. 9. when there is data with nodes

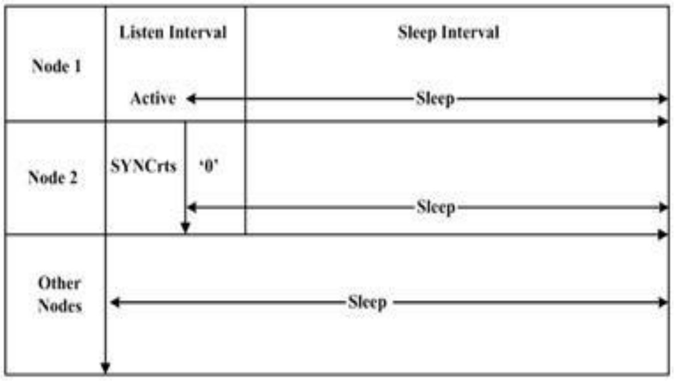

Fig. 10. When no data with nodes

3) Acknowledgment

Both S-MAC and TEEM protocols send ACK packet after each fragment received by receiver. They do so to avoid collision at receiver due to hidden terminal problem. Through ACK packet both protocols ensure receiver neighbors that medium is still busy. But it wastes energy too, due to overhead. ETEEM protocol sends one ACK packet for the whole burst i.e. for all fragments. It minimizes energy consumption because of sending ACK packet per whole burst. After missing packet received by the receiver, it acknowledges whole burst. There is possibility of collision at receiver. ETEEM compromises one collision per communication but has very little chances to occur as experimental result shows. If it occurs, it still consumes less energy from sending ACKs per fragment. On one collision, less energy is consumed and only one packet will be lost which can be recovered later. Collision can be minimized as in S-MAC and TEEM protocols but penalty will be paid in the form of energy consumption.

4) Algorithm for ETEEM

ETEEM will follow the work flow and algorithm as shown in figure 11. The figure shows that if sender wants to send data to receiver it first sends SYN Crts packet after receiving CTS both parties establishes connection. After last packet received with FLAG 0 it sends missing fragment sequence number to the sender. Sender will send again missing packets. If last packet is received and FLAG is 1 the receiver will first send lost packet sequence number and then sends CTS packet again. The algorithm will start working again from point 4.

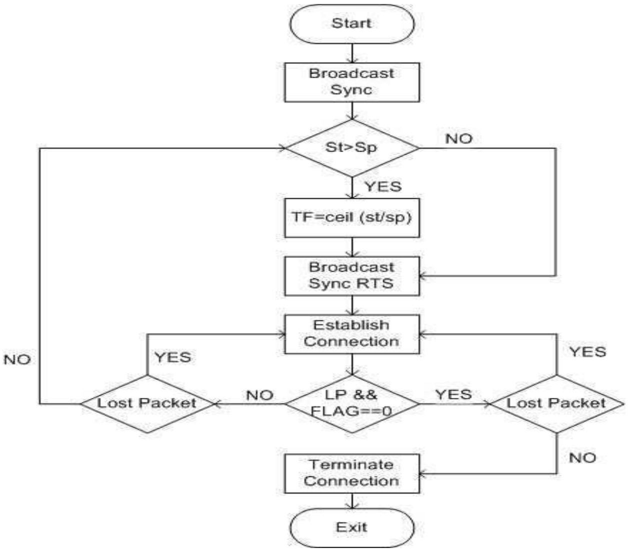

Fig. 11. Workflow of ETEEM protocol

IV. Performance Analysis

We took 03 nodes and simulation time as 50 seconds but it can be changed according to needs. Further, for SYNC packet and control packets, we took real values of energy consumption from the simulation of S-MAC and TEEM. Assumed that each node can send and receive 10 bursts of data and can change as per need. We implemented S-MAC, TEEM and ETEEM for three possible conditions i.e best case, average case and Worst case and calculated results. In order to evaluate these cases we make data as constant for all these protocols.

We divided our experiment in two parts: in first part, we devised formula for these protocols based on their algorithms and working was implemented for all possible conditions i.e. for Best Case, Average Case and Worst case. Experiment shows that ETEEM consumes less energy as compared to S-MAC and TEEM. S-MAC was the first protocol to implement in WSN.

1) S-MAC Protocol

We first took single node and calculated algorithm of S-MAC and then implemented that calculation on N number of nodes. For topology creation and further contention for the medium Node first sends SYNC packet and Control Packets, therefore we have as in equation (1) below

$$N = SYNC + ControlPacket \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where ControlPacket = RTS + CTS hence equation (1) becomes

$$N = SYNC + RTS + CTS \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

After successful transmission of control packets, data transmission will occur, so which is generating the same results as of equation (5)

$$N = SYNC + RTS + CTS + Data \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

As we know that Data = ACK + Fragment because after each fragment S-MAC also acknowledges. Hence equation (3) becomes

$$N = SYNC + RTS + CTS + ACK + Fragement \ldots\ldots (4)$$

As Data is variable, one node can send as many data fragment as it has in its buffer and after each fragment it also receives an ACK. After putting this scenario in equation (4)

$$N = SYNC + RTS + CTS + \sum_{x=1}^{n}(ACK + Data) \ldots\ldots\ldots (5)$$

Similarly for total number of nodes N_t the operation will be described as follows

$$N_t = \sum_{y=1}^{n}(SYNC + RTS + CTS) + \sum_{x=1}^{N}(ACK + Data) \ldots (6)$$

2) TEEM Protocol

TEEM protocol is modified version of S-MA C i.e. SYNCrts in place of SYNC. In order to devise formula for TEEM protocol we will have to first look to single node and then will implement it on N number of Nodes. For single node

$$N = SYNCrts + CTS + \sum_{x=1}^{N}(ACK + Data) \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

For n number of nodes the formula will be like following

$$N_t = \sum_{y=1}^{n}(SYNCrts + CTS) + \sum_{x=1}^{N}(ACK + Data) \ldots\ldots\ldots (8)$$

3) ETEEM Protocol

As ETEEM is modified form of TEEM protocol, hence it uses FLAG as extra packet and uses one ACK for whole burst of data. For topology generation it also uses SYNCrts packet. For single node algorithm of TEEM protocol:

$$N = SYNCrts + CTS + \sum_{x=1}^{n}(ACK + Data + FLAG) \ldots (9)$$

For total number of Nodes Nt, it will be:

$$N_t = \sum_{y=1}^{n}(SYNCrts + CTS) + \sum_{x=1}^{n}(ACK + Data + EOD) \; (10)$$

V. Performance Cases

1) Case 1

In Case 1, ETEEM protocol is compared with S-MAC and TEEM protocol for the scenario in which all these protocols consume their energy up to their high level as its algorithm support. We make a congested environment and force these protocols to collide each and every fragment sent by any node to its neighbor. We observed that nodes are active for their whole life cycle. For each fragment loss, S-MAC and TEEM protocols send ACKs twice i.e. one negative and one positive ACKs. ETEEM protocol sends only two ACKs i.e. one negative ACKs and one positive per burst. Experiment are depicted in figure 12 which shows that ETEEM consumes 0.0035% and 0.0043% less energy from TEEM and S-MAC respectively

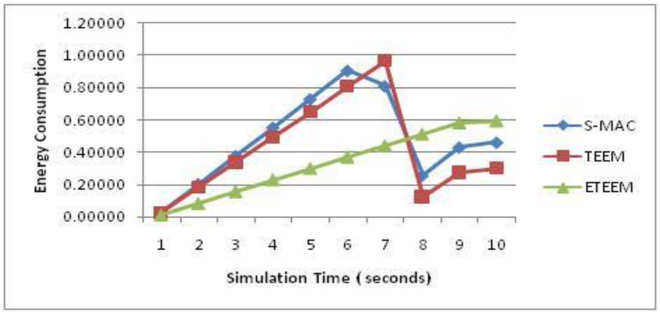

Fig. 12. Case 1 Scenario

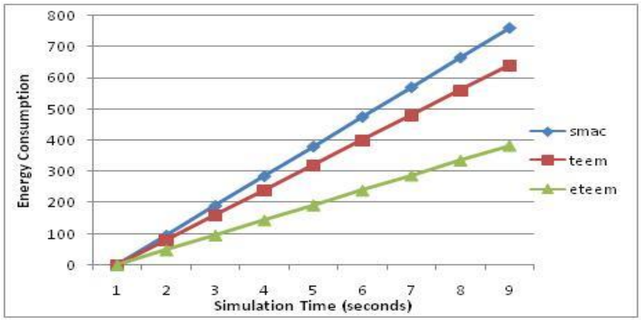

Fig. 13. Case 2 Scenario

2) Case 2

Case 2 is the scenario in which there is average traffic going on. We allowed large amount of data to be transferred in such a way that data transfer can occur in sleep quantum. All protocols are active for specified time equally depending on the data. But as S-MAC uses SYNC and RTS packets separately therefore it consumes more energy. TEEM combined these two packets and save some part of the energy but like S-MAC it acknowledges each and every data fragment. In collision S-MAC and TEEM almost send ACKs twice per collide fragment as compared to their normal routine. The performance of ETEEM is observed better than these protocols because despite the fact that it is active for the time as TEEM is, yet it sends only two ACKs i.e. one negative ACK and one positive, hence saves more energy than S-MAC and TEEM. Figure 13 shows that in average case ETEEM consumes 0.0032% and 37.5% less energy as compared to TEEM protocol and S-MAC respectively.

3) Case 3

Case 3 is the scenario in which it consumes minimal energy. As S-MAC uses fixed interval of time for Listen and Sleep schedule. Therefore Case 3 of S-MAC will be to go to sleep state within given schedule. Similarly for TEEM will be to go to sleep state in SYNCnodata part of Listen interval i.e. go early to sleep state. For ETEEM it will be the one in which it send FLAG packet with value 0 and go to sleep state early then its normal schedule. Best case result of these protocols are depicted below in figure 14 which shows that in best case ETEEM consume 70% less energy than TEEM protocol and subsequently consumes 92.5% less energy.

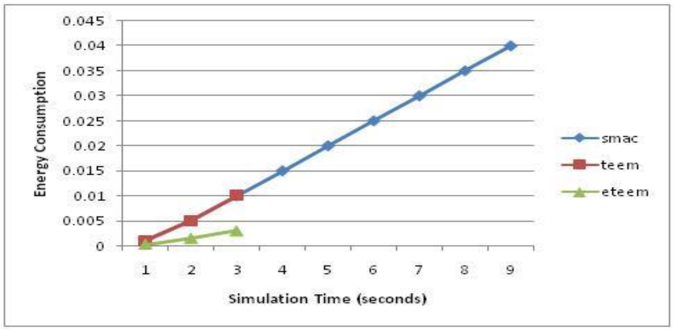

Fig. 14. Case 3 Scenario

VI. Conclusions and Future Work

This paper presents new MAC protocol for WSNs called ETEEM. It has very good energy conserving mechanism as compared to S-MAC and TEEM. Experimental result shows that it consumes less energy in all three possible cases (best case, average case and worst case). ETEEM protocol uses FLAG packet instead of SYNCnodata or SYNCdata part of TEEM. It uses one ACK per whole burst for the sake of energy conservation. For experimental purpose C language is used. Energy comparison is showed for three possible cases i.e. Best case, Average case and worst case. ETEEM consumes less energy in all three cases discussed in previous section. Future work includes experimentations of this protocol in different simulations tools. Nodes and Data can be varied to examine optimum results.

References

[1] Akyildiz, Ian F., et al.”Wireless sensor networks: a sur vey.” COMPUT NETW 38.4, 2002: 393-422.

[2] Chong, Chee-Yee, and Srikanta P. Kumar. ”Sensor networks: evolution, opportunities, and challenges.” Proceeding s of the IEEE 91.8, 2003: 1247-1256.

[3] Akyildiz, Ian F., Dario Pompili, and Tommaso Melodia.” Underwater acoustic sensor networks: research challenges.” Ad hoc networks 3.3, 2005: 257-279.

[4] Polastre, Joseph, Robert Szewczyk, and David Culler. ” Telos: enabling ultra-low power wireless research.” LECT NO TES COMPUT SC, Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on. IEEE, 2005.

[5] Javaid, N., Ahmad, A., Rahim, A., Khan, Z. A., Ishfaq, M., and Qasim, U. (2014). Adaptive medium access control protocol for wireless body area networks. INT J DISTRIB SENS N, 2014.

[6] Alkhatib, Ahmad Abed Alhameed, and Gurvinder Singh Baicher. ”Wireless sensor network architecture.” International conference on computer networks and communication systems (CNCS 2012) Vol. 35. 2012.

[7] Philipp Sommer,Roger Wattenhofer Lars Schor, ”Towards a zero-configuration wireless sensor network architecture for smart buildings,” in First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, 2009, pp. 31-36.

[8] Lu, Chenyang, et al. ”Rap: A real-time communication arc hitecture for large-scale wireless sensor networks.” Real -Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE. 2002.

[9] Anastasi, Giuseppe, et al. ”Energy conservation in wire less sensor networks: A survey.” Ad Hoc Networks 7.3, 2009: 537-568.

[10] Ye, Wei, John Heidemann, and Deborah Estrin. ”An energy -efficient MAC protocol for wireless sensor networks.” IEEE INFOCOM SER 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. Vol. 3. 2002.

[11] Koen Langendoen Tijs van Dam, ”An adaptive energy efficient MAC protocol for wireless sensor networks,” ACM, pp. 171 - 180, 2003.

[12] Du, Shu, Amit Kumar Saha, and David B. Johnson.” RMAC: A r outing-enhanced duty-cycle MAC protocol for wireless sensor networks.” IEEE INFOCOM SER 2007. 26th IEEE International Conference on Computer Communications. IEEE. IEEE, 2007.

[13] Changsu Suh and Young-Bae Ko, ”A traffic aware energy effi cient MAC protocol,” IEEE, vol. 3, pp. 2975-2978, May 2005.

[14] Crow, Brian P., et al. ”IEEE 802.11 wireless local area networks.” Communications Magazine, IEEE 35.9, 1997: 116-1 26.

[15] Eusthatia Ziouva, ”CSMA/CA performance under high tra ffic condition: throughput and delay analysis,” computer communication, vol. 25, pp. 313-321, May 2002.