Congestion-less Energy Aware Token Based MAC Protocol Integrated with Sleep Scheduling for Wireless Sensor Networks

Soumya Ray, Subhasis Dash, Nachiketa Tarasia, Anuja Ajay, Amulya Ratan Swain

***Abstract*-Technological advances in microelectronic and telecommunication fields enable the existence of tiny computing units, very small and more efficient. In the context of these new technologies, Wireless Sensor Networks (WSN) have emerged. The main research objective in WSN domain is the development of algorithms and protocols ensuring minimal energy consumption. In this paper, we propose on energy efficient token based MAC protocol integrated with sleep scheduling for WSNs, in order to reduce energy consumption of each sensor node which is one of the important issues to prolong the network lifetime. To derogate energy consumption most of the MAC protocols in WSN exploits low duty-cycle; among those RMAC, HEMAC allows a node to transmit data packet for multi-hop in a single duty-cycle. To reduce energy consumption on prolonged network life time sensor networks are usually duty cycled; each node remains in low power sleep mode most of the time and wakes up periodically to sense for channel activities. In the above said MAC protocols, due to the synchronized scheduling and transmission collisions, flooding increases resulting in energy waste and low throughput. Allowing for nodes to operate with a new sleep based token approach; we intend to produce energy efficiency in an event based approach by cutting down flooding, collision and traffic congestion. Simulation studies of the proposed MAC protocol have been carried out using Castalia simulator.**

***Keywords* — Energy consumption, MAC Protocol, Sleep Scheduling, Token, Traffic Congestion.**

I. Introduction

WSN are composed by tiny sensors called sensor nodes. A sensor node ensures main functions like acquisition of environment physical measures, local processing of collected data and their transmission to the sink. These function are performed by an electronically modules composing the node's hardware architecture. Other components can be added to ensure mobility function and energy self-recharging. A node is alimented by a limited quantity energy (battery) and communications wirelessly using a radio signal. Generally, communication between nodes consumes more energy than local processing or collecting data operation. The geographical nature of the deployment space of nodes makes quasi impossible the replacement or the recharging operations of batteries. The challenge is to economize energy inside every node in order to maintain as long as possible the network functionality.

An event is defined as the critical data generated by sensor nodes that should be detected reliably and communicated to the sink. Reliable event detection enables the transfer of critical data to the sink by adjusting the reporting frequency to detect events with low latency and data accuracy quickly [13].

With objective of prolong the life time of WSN, reducing energy consumption turns out to be the most crucial factors for almost all the WSNs protocol exploring, particularly for the MAC protocol that directly insures the state of the main energy consumption component i.e. the Radio module [8]. MAC layer is considered as an important source of energy wastage that we summarize as follows [1, 2]:-

Overhearing: A sensor node receives packets that are transmitted for other nodes, which forcing every node of the neighbourhood to waste energy when receiving this radio.

Collisions: Since radio channel is shared by many nodes, a collision take place every time when two nodes try to send in the same time their packets. Collisions increase energy consumption and latency packets deliverance mechanism due to retransmissions.

Control packets (overhead): packet headers and control packets (RTS/CTS/ACK) used by a MAC protocol do not contain application data, thus they are considered as supplementary data (overhead).

Idle listening: when a node is not active, leaves listening the signal carrier to knowing if it is the receiver of an eventually traffic. In this situation, the amount of energy waste is equal to the energy dissipated by a normal reception.

In order to decrease or if possible to eliminate these various sources of energy wastage, several protocols has been proposed these last years and which are divided into two main classes:

1. *Scheduled-based protocols*: These protocols known as deterministic are employed to avoid collisions by associating a slot time for each sensor node in a given cluster, and to mitigate the effects of overhearing problem, because in this situation each node knows his corresponding slot time to transmit data packet [3].

2. *Contention-based protocols:* These protocols known as CSMA-based are usually used in the multi-hop wireless networking due to their simplicity and their adequacy to be implemented in a decentralized environment like WSN. To decrease collisions and to reduce considerably other sources of energy wastage, the Wake-up/Sleep mechanisms and/or the control messages RTS/CTS/ACK defined in 802.11x standard, are used to design energy efficient MAC protocols for WSN like S-MAC, T-MAC, B-MAC and Z-MAC.

Manuscript received March 6, 2011; revised April 8, 2011.

Soumya Ray is with the School of Computer Engineering, KIIT University, Bhubaneswar, Orissa PIN 751024 India (phone: 9861241404; e-mail: somi.gita@gmail.com).

Subhasis Dash is with the School of Computer Engineering, KIIT University, Bhubaneswar, Orissa PIN 751024 India (e-mail: subhasisbbsr@gmail.com).

Nachiketa Tarasia is with the School of Computer Engineering, KIIT University, Bhubaneswar, Orissa PIN 751024 India (e-mail: ntarasia06@gmail.com).

Anuja Ajay is with the School of Computer Engineering, KIIT University, Bhubaneswar, Orissa PIN 751024 India (e-mail: anujaajay@gmail.com).

Amulya Ratan Swain is with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore PIN 560012, India (e-mail: swainamulya@gmail.com).

In this paper, by allowing nodes to operate with a new token-based approach, we intend to produce energy efficiency in an event based environment by reducing flooding, collision and traffic congestion. Here, duty cycle that will decide which all nodes would receive data packets along with defining the sequence of reception at the synchronization period (At one time only one node will be able to communicate with the sink). It would keep the node sequence in a queue at the sink and the data packets will be transmitted sequentially. Since the hop sequence already exists at the sink, the nodes would stay active for a much lesser period to acquire the token and would continue in sleep for the rest of the span which would eventually turn it energy efficient mitigating the wastage of energy. Please note that, we are employing two keys in this protocol: one is Token and the other being Queue in order to schedule the nodes in packet transmission. Our performance evaluation shows that Token-Based MAC protocol integrated with sleep scheduling can achieve better energy-throughput tradeoffs and extend node life span substantially while providing fewer collisions. Implementing token based MAC allows only a single source to interact with the sink, thus cutting down congestion and ensuring reliable data delivery.

The rest of this paper is organized as follows. In section II, we give a survey of related works. In section III, we describe a proposed MAC protocol and comparative that of SMAC. In section IV, result of performance evaluation of the proposed protocol, SMAC, RMAC and their comparisons are given. In section V, we draw a conclusion.

II. RELATED WORK

Due to limited energy in WSN energy efficiency is an important factor in designing a MAC protocol. Energy management is concerned of which set of nodes should be turned on/off and when, for the purpose of energy saving and network longevity. Several Medium Access Control (MAC) protocols have been proposed to reduce the energy consumption by sleep planning methods in order to increase the lifetime of sensor networks. However, they lead to a large increase in end-to-end latency, which affects the efficiency of the network. In some of our previous work we have improved the lifetime without increasing the latency. Traffic-Adaptive MAC (TRAMA) is a scheduling protocol that determines which node can transmit at a particular slot based on the traffic information. However, Time Domain Medium Access (TDMA) protocols are complex to maintain in a multi-hop network due to their timing synchronization. They are not suited for event-based operation, as they cannot increase the resource utilization due to their reservation schemes. IEEE 802.11 distributed coordination function (DCF) is a Carrier Sense Multiple Access CSMA type protocol in which energy consumption is very high due to idle listening of nodes.[13] Sensor-MAC (S-MAC) is a contention-based protocol with integrated low-duty-cycle operation that supports multi-hop operation. S-MAC puts all nodes into periodic listen and sleep. Even though energy saving is higher in S-MAC, periodic sleeping increases latency, since a sender must wait for the receiver to wake up before it can send out the data. Timeout MAC (T-MAC) reduces idle listening by transmitting all messages in bursts of variable length and sleeps in the remaining time. However, the latency in T-MAC increases because data arrived during sleep is queued until the next active cycle. In all these above MAC protocols, due to the synchronized scheduling and transmission collisions, flooding increases resulting in energy waste and low throughput.

Performance studies show that while wake-up schedules are effective in reducing energy consumption in sensor networks due to the sporadic characteristics of sensor traffic, the delay incurred by waiting for the next of forwarding node to be awake viz. Sleep latency can be quite large. The wake-up schedule is a key component in the design of a duty-cycle MAC to mitigate energy consumption. In SMAC [5], TMAC [5], RMAC [6] and HEMAC [7] require synchronization among nodes, which can be complex and expensive particularly in large multi-hop networks which clock drifts, low duty-cycles and transient link qualities.

III. TOKEN BASED MAC PROTOCOL

Token based MAC protocol is a MAC protocol that employs token information to transmit data for multi-hops in a single duty-cycle. It uses a Request message to synchronize all the required source nodes (remaining nodes are in sleep mode) to relay the upcoming data. Unlike SMAC, in which all nodes have the same synchronized and periodic listen and sleep cycle, in our protocol different synch format is exercised. In Token-based MAC protocol, each node stores its node ID (Ni), Parent node (Pi). Sensor node other than sink node has an array to store data packets, and stores the reply packets sent through it. Sink has a queue to store the request packets. Sink has a token that is used for synchronisation.

The token based MAC protocol is divided into 2 phases:

- Level discover phase(Listen phase)
- Synchronization with sleep scheduling (Request & token allotment in Listen phase) & Data Transmission phase(the nodes which are not participate in data transmission they remain in sleep mode)

A. *Level discover Phase:-*

The Sink sends out level discovery (ADVT) packets and all the neighbouring nodes that receive these packets assign themselves with level 1. The nodes uphold to send out packets that include their level number and identities until all the nodes in the network are assigned with a level. The sensor nodes assign their levels according to the hop distance from the sink node to a source node. A node is said to be in level ‘L’, if it is ‘L’ hops apart from the sink node. The sink is a level ‘0’ node. The level ‘N’ nodes have the path length of ‘N’ hops back to the sink. Once the nodes are deployed, the sink broadcasts the ADVT packet in order to discover the level of all the nodes and sets its parents in the respective manner. Post an ADVT message is transmitted by sink node the hop count records how many hops it has travelled from the sink. The hop count is increased by one each time when a node receives the ADVT message. While receiving an ADVT message a node considers itself in level ‘N+1’ if the hop count received is ‘N’. If a smaller hop count ADVT message is received, the node updates its level according to the new hop count. The parameters of ADVT message are ‘Ni’, ‘HCi’. Thus the ADVT message is used to model the network in to levels and implement a path from each sensor node to the sink.

B. *Synchronization(sleep scheduling) & Data transmission Phase*

The Sink is allotted with a token. During the synchronization period, all those sensor nodes that desire to send the data to the sink will send out Request packet to the sink node. The parameters of request message are node ID of the node that is requesting the token, its parent ID. If a source node requires sending a data packet to the sink node then the source node would request to its parent node and this process will go on until the request message arrives at the sink node. There exists a queue at the sink node that keeps track of the request messages according to timestamp of the source node in an ascending order. In case if there happens a scenario when multiple nodes at once send out these request packet to acquire token from the sink, the request messages will be staged in an FCFS (first come first serve) fashion. Post receiving the request message the sink node will allow the token to pass through the respective child nodes to the desired source node using a reply packet. The parameters of reply message are node ID of the node that requested the token, token of sink, and node ID of child node that forwarded the request. We are using sleep schedule mechanism to reduce energy consumption where we use a flag value in a reply packet. While passing through the node the Reply packet sets the flag value to 1. The nodes that have not participated in the message passing set the flag value to 0 and go to sleep mode. Since we are using event based message passing technique, when an event takes place in a sleep period of a node the node keeps the event message and maintains its request packet in an array. When the node wakes up it sends the remaining request message in the array at sleep period.

The node that stands at the first place of the queue gets the token from the sink. Post receiving the reply packet containing the token, the node transmits data packets. Parameters of data packet are parent ID, token returned or not, and data. Every source node has got a queue to uphold numbers of packet transmission. Until the token is there with the source node it can continue with the data transmission to the sink node. Once the data transmission concludes the token would be sent back to the sink node with the last data packet transmitted from the source. The release of the token grants permission to the next source node to have the data packet transmission and the process continues so on. After the sink receives the token it would send out reply packet, containing the token, corresponding to the next request in the queue.

The node will supply its own ID on the required node and send out to the Sink ID via its parent node. When the sink S will send out the token in the reply packet to the required node, the packet will be backtracked through the path traversed by the request.

For example, when an event takes place, the queue will hold Request message of all the sensor nodes that requested the token. In the above context, the node 4 has transmitted the Request message. The Sink checks out the queue and then sends the token to node 4 by employing the Reply message. Then 4 will get the token. Each node has its own queue for data packet transmission. If node 4 has 7 data packets then it will maintain the data packets sequentially in its queue in an order of 1 2 3 4 5 6 7. Now that the token is available at the node 4, hence it will send out data packets one by one to the sink. When the last packet will be transmitted to the Sink, the token would be attached with this last packet and be released to the Sink. The next node sequence i.e. node 6 will get the token as node 4 in the first case. This process will move on until all the nodes in the queue finish data packet transmission. When there is data transmission between node 4 and sink rest all nodes go to sleep mode. Only the path (nodes) that is required for data transmission remains in sleep mode. If an event takes place in a sleep mode at node 9 then node 9 creates an array and keeps the message for data or event in the array as a request message. Once sleep mode changes to wake up mode, node 9 sends the message to the sink.

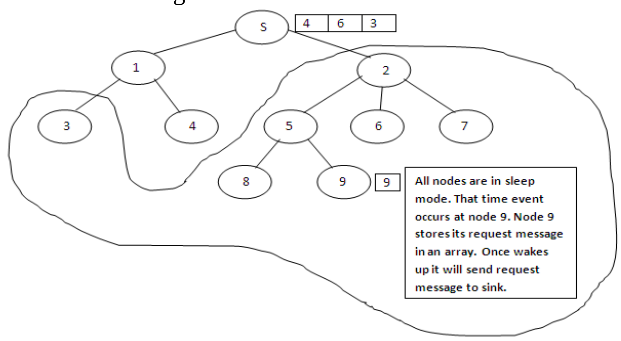

Fig1.Request packet en-queue at SINK node and the nodes in the closed area are lying in Sleep mode .

Acknowledgement packets are associated with reply and data packets to ensure that data packets and reply packet with tokens are not lost.

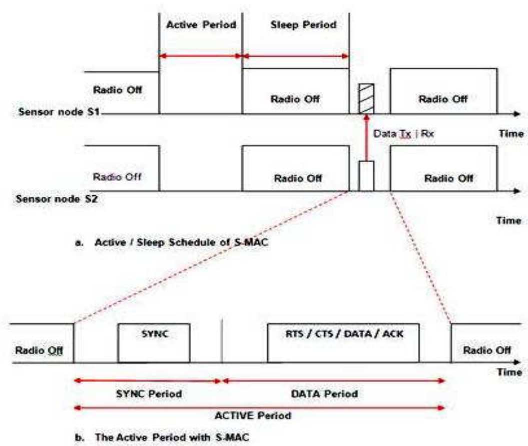

Fig.2: Active sleep scheduling in SMAC

In SMAC, sensor nodes employ in a continuously alternate sleep and active period. During the sleep period, nodes switch their radio off and hence save energy. Synchronization as well as frame transmissions and receptions are performed during the active period by using a contention based scheme. But Token-Based MAC protocol has no fixed duty cycle. It will only be active when an event takes place. For the rest of the life span it lies in the sleep period. The same is the reason that it will save more energy than SMAC. The active period in SMAC is divided into two consecutive phases: the Synchronization period followed by the Data period. In token-based MAC protocol, active period is divided into two consecutive phases: Synchronization period followed by the Listen period. In synch period they get synch RTS message and listen period they make queue for all required nodes that send synch RTS message. The data will be transmitted in the sleep period.

In case of SMAC data transmission is an end to end delivery whereas in this case in Token-Based MAC data transmission is in hop to hop delivery fashion. In Token-Based MAC data is transmitted in more than 2 hops but at

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

once only one node will interact with SINK, the node who has got the token with it, which avoids collision in the data packet transmission.

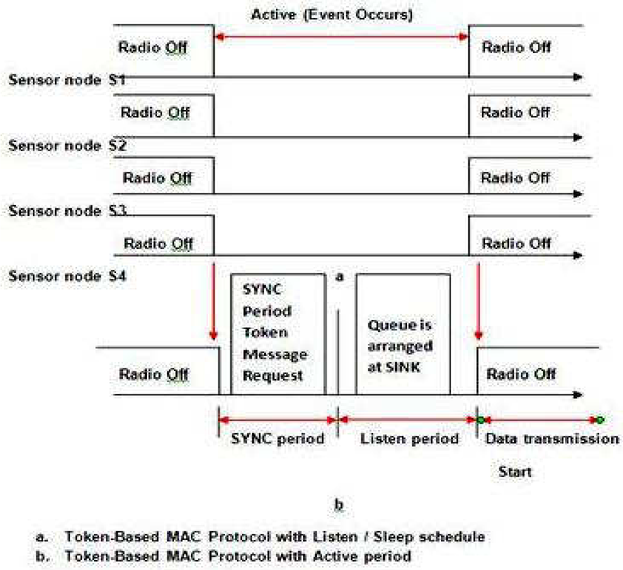

Fig.3: Listen sleep scheduling Token bases MAC

We make some assumptions for surveillance applications in WSNs, which would be used as points of reference frame in the further studies. WSNs comprise of a number of sensing nodes which are distributed in a wide area, according to the demand of the application. The base station (sink), which accumulates data from other nodes, interacts with a user (someone interested in monitoring the activity). Sinks have more advanced features than sensing nodes in terms of data transmission and processing capabilities, memory size and energy reserves. There can be multiple sinks for a network in order to avoid any single point of failure. Energy dissipation is a major factor in WSNs during communication among the nodes. Energy should be preserved, so that the batteries do not get drained quickly as these are not easily replaceable in a typical surveillance scenario. QoS tries to ensure efficient communication within bounded delays. Protocols should check for network stability and redundant data can be transmitted to gain reliability. It is also necessary to maintain certain resource limiting factors, such as bandwidth, memory buffer size and processing capabilities. The transmission mode plays a significant role in WSNs. Nodes can take a single-hop or multi- hop path depending upon the type of network topology chosen for transmitting data to other nodes in the network. The sensor nodes can be mobile or static depending on the application. In surveillance applications, sensor nodes are often placed in unattended areas. Therefore, the network should be self organizing and self-creating.

C. Proposed Algorithm

Algorithm executed at each sensor node N_j on receiving a packet from node N_i

Phase 1: *Level discover phase*

/*On receiving ADVT packet*/
  if ADVT packet **then**
    if(HC_j==∞)
      $HC_j=HC_i+1$;
      $P_j=N_i$;
      Broadcast ADVT(N_j,HC_j);
    else if((P_j not sink) and ($HC_j>HC_i+1$))
      $HC_j=HC_i+1$;
      $P_j=N_i$;
      Broadcast ADVT(N_j,HC_j);
    else
      Discard the ADVT;
    end if
  end if

Phase 2: *Synchronization (sleep scheduling) & Data phase*

/* On receiving Request packet from node N_i to node N_j*/
if Request packet **then**
  if (Pi==Nj) **then**
    Broadcast Request msg(RequiredNodeID ,Pj) ;
    Flag==1;

  else if (Ni== sleepmode**)**
    Put request packet in array Ai;
  end if
end if
/*On receiving Reply message*/
if Reply packet then
  if (Ni==Pj && Nj!= RequiredNodeID) **then**
    Broadcast reply packet;
    Put reply packet in Reply array;
    Set Timer1;
    Broadcast Ack packet to Pj for Reply packet;
  else if (Ni==Pj && Nj==RequiredNodeID && token == true) **then**
    TokenNode=True;
    Set timer3;
    If last data packet to send **then**
      TokenNode=false;
      Flag = 0;
    end if
  Broadcast Ack to Pj for Reply packet;
  Broadcast Datapacket(Data, TokenNode,Pj);
  Put data packet in Data array;
  Set Timer2;
  end if
end if
/*On receiving Data packet*/
if Data Packet **then**
  if (Pi==Nj) **then**
    Broadcast Datapacket (Pj, Token node, data, send);
    Set Timer2;
  end if
end if
/*Timer1 timeout */
if Timeout **then**
  if (any reply packet in the array)
    Retransmit it;
    Set the Timer1;
  end if
end if
/*Timer2 timeout */
if Timeout **then**
  if (any data packet in the array)
    Retransmit it;

```
                Set the Timer2;
          end if
    end if
    /*Timer3 timeout */
          if Timeout then
          if(flag==0 && !=Sink)
          set radio =sleep;
           set timer4;
          end if
    end if
    /*Timer4 timeout */
          if Timeout then
          set radio =listen;
       if (request packets in array ) then
           Broadcast Request msg(RequiredNodeID ,Pj) ;
      end if
    end if
```

ALGORITHM EXECUTED AT SINK NODE

```
/*On receiving Request message*/
If (Request packet) then
      Queue Request message;
      if (Token==true)
            Pop Request message;
            Broadcast Reply message;
            Set Timer1;
            Token = False;
      end if
end if

/* On receiving Data packet*/
if (Data packet) then
      if(tokenNode==false)
            Pop Request message;
            Broadcast Reply message;
            Set Timer1;
            Token = False;
      else
            Broadcast Ack for data packet;
      end if
 end if
/*Timer1 timeout */
if Timeout then
      if ( any reply packet in the array )
            Retransmit it;
            Set the Timer1;
      end if
end if
```

IV. SIMULATION EVALUATION

In order to evaluate our Token-based MAC design, we used Castalia simulator [10]. We compare the Token based MAC against SMAC and RMAC. In SMAC we are using end-to-end delivery while in RMAC and Token-based MAC protocol hop-to-hop delivery is used. For this simulation, the network parameters, such as transmission range, transmission rate, sensitivity, transmission power etc., are similar to the parameters specified in CC2420 [11] data sheet and TelosB [12] data sheet.

All the nodes in the network have already been synchronized to use a single wake up and sleep schedule. The synchronization is done in level phase. Nodes will wake up at the beginning of the synchronization period and listen to the medium. We will use a reserved path by the help of request message for data transmission which assures the shortest path between any two nodes.

A. Path length evaluation

In Fig 4 it shows the histogram of the path lengths from the sensors to the sink. The maximum path length from a sensor to the sink is 5 hops, but in RMAC we find it 15 hops. If hops number will decrease, then automatically energy consumption will be decreased.

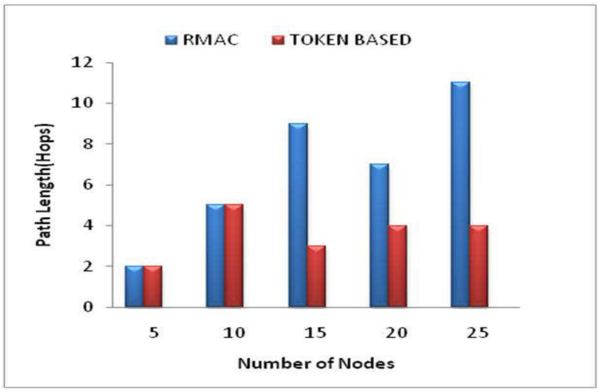

Fig 4: The Histogram of the path lengths of the realistic network

B. Energy Consumption Evaluation

In Fig 5 it also demonstrates the impact that traffic contention has on energy efficiency. It shows the power consumption with respect to number of packets transmission. We evaluate the energy efficiency of Token-based MAC .We have varied our traffic load up to 100 packets in each topology and then observed the sensor power consumption during the entire simulated time. Each simulation runs for 3000 seconds of simulated time. Fig 5 shows the average power over all the sensors in this scenario. Average power consumed by the sensors by the total simulated time. Error bars show the minimum and the maximum values for a single sensor's average power consumption. As the traffic load increases, both RMAC and SMAC increase their energy consumption, but Token-based MAC has a smaller rate of increase than both SMAC and RMAC.

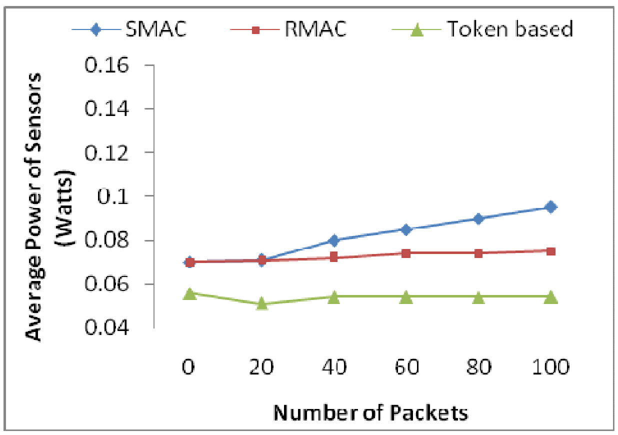

Fig 5: Average power of sensors

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In fig 6 Energy consumption and average end-to-end latency is analysed for S-MAC and Token-Based MAC protocols, when the number of nodes is varied. We assume the average hop distance between the event–sink is 5.In S-MAC; energy consumption is high as there is periodic sleep of the nodes. Energy consumption in Token-Based MAC is less because the event-sink paths are kept active only for the event duration by the dynamic schedule.

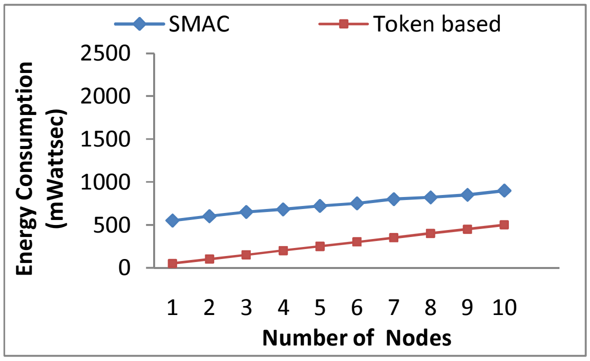

Fig 6: Energy Consumption Vs Number of Nodes

C. Latency evaluation

For a multi-hop delivery of a packet, sensors in Token-based MAC transmit packets in minimum hops than RMAC. Flooding is avoided which further increases the energy efficiency of the entire network with Token-based MAC. Another reason for the Token-based MAC being more energy efficient is that sensors in Token-based MAC never consume energy on overhearing a data frame transmission, because during data frame transmission, all the nodes are in the sleep mode except the nodes on the reserved path towards the sink.

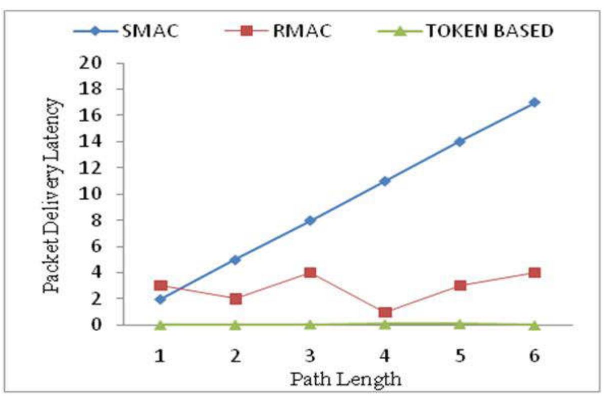

Fig 7: Delivery Latency

We evaluate the performance of hop-to-hop delivery latency. We use typical light traffic load for sensor networks. Each simulation runs for 3000 seconds of simulated time. Here we are using token request message. It consumes some time after that also its packet delay time is less than SMAC and RMAC.

V. CONCLUSION AND FUTURE WORK

Token-based MAC mechanisms have been used in sensor networks in order to improve the energy efficiency along with data accuracy, but they also introduce significant increase in hop-to-hop delivery latency and idle listening as well. We have presented the design and evaluation of Token based MAC as a duty cycle MAC protocol that is capable of multi hop data delivery in a single operational cycle. We have also introduced the data reliability using dynamic sleep schedule, which activates the corresponding nodes in the source-sink path. Our sleep schedule reduces the energy consumption by activating the nodes only in the event area and controls the traffic with low latency and low energy consumption. We are using a token to select a single sensor node that communicates with sink at a single time and will avoid the flooding in the network. Our simulation evaluation shows Token-based MAC's advantages in reducing delivery latency and energy consumption in the network.

Theoretical analysis of Token-based MAC could guide us in the future exploration. Eventually, the token mechanism reduces the energy and decrease the complexity in packet handling. In order to increase the energy efficiency and decrease the delivery latency we can explore the token mechanism in binary tree network which may reduce the latency and energy consumption.

REFERENCES

[1] Zhi-Wen, O., Shruthi, B. K. and Sang, S. K. (2005) Medium Access Control for Wireless Sensor Networks. CS258 - Advanced Communication Networks, San Jose State University.

[2] Injong, R., Ajit, W., Mahesh, A. and Jeongki, M. (2005) Z- MAC: An Hybrid MAC for Wireless Sensor Networks. Dept. of Computer Science, North Carolina State University, SenSys'05, November 2–4.

[3] Sungrae, C., Kanuri, K., Jin-Woong, C., Jang-Yeon, L. and June, S.-D. (2005) Dynamic Energy Efficient TDMA-based MAC Protocol forWireless Sensor Networks. Autonomic and Autonomous Systems and International Conference on Networking and Services, ICAS-ICNS, Vol , Issue , 23-28, P.48 - 48

[4] Binu. G. S and K. Poulose Jacob**, "**A Channel Adaptive MAC Protocol with Traffic Aware Distributed Power Management in Wireless Sensor Networks", *InternationalJournal on Computer Science and Engineering IJCSE*, V2,(2), 2010, pp. 177-182.

[5] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," Proc. of INFOCOM ,NewYork:IEEE, no. 3,pp. 1567-1576, 2002.

[6] S. Du, A. Saha and D. Jhonson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," Proc .INFOCOM, pp. 1478-1486, May. 2007.

[7] Kyong-Tak Cho, and Saewoong Bahk, "HE-MAC: Hop Extended MAC Protocol for Wireless Sensor Networks," in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE *on*, Nov. 30 2009-Dec. 4 2009, pp. 1–6.

[8] G. Zheng, Y. Gao, J. Fu, S. Tang , "An Energy Efficient and Low Latency MAC Protocol for Multi-hop Wireless Sensor Networks," International Conference on Communications and Mobile Computing, Vol. 3, pp. 224-228, 2010.

[9] W. Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACM Transactions on Networking, Vol. 12, no. 3,pp. 493-506, Jun. 2004.

[10] "Castalia a simulator for wireless sensor networks," http://castalia. npc .nicta.com.au/ pdfs/ Castalia User Manual.pdf.

[11] "Cc2420 data sheet," http://www.stanford.edu/ class/ cs244e/ papers/ cc2420.pdf.

[12] "Telosb data sheet," http://www.xbow.com/Products/ Product pdf files/ Wireless pdf/ TelosB Datasheet.pdf.

[13] N.A Vasanthi and S.Annadurai ,"Sleep Schedule based on Reliable Event Detection For Wireless Sensor-Actor Networks,"ICGST-CNIR journal,volume 7,issue 1,July 2007

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)