ISSN:0975-9646

Soumya Ray et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (2) , 2011, 747-753

Energy Efficient Token Based MAC Protocol for Wireless Sensor Networks

Soumya Ray[#], ,Subhasis Dash[#] ,Nachiketa Tarasia[#] ,Anuja Ajay[#], Amulya Ratan Swain[*]

[#]*School of Computer Science Engineering, KIIT University*
Bhubaneswar,India

[*]*Department of Computer Science and Automation*
Indian Institute of Science
Bangalore,India

***Abstract*– In this paper, we propose on energy efficient token based MAC protocol for WSNs, in order to reduce energy consumption of each sensor node which is one of the important issues to prolong the network lifetime. WSN consists of a large number of small and low-cost sensor nodes powered by small batteries and equipped with various sensing devices. Due to limited energy in WSN energy efficiency is an important factor in designing a MAC protocol. 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 all these above MAC protocols, due to the synchronized scheduling, transmission collisions, flooding will increase resulting in energy waste and low throughput. By allowing nodes to operate with a new token-based approach, we intend to produce energy efficiency in an event based approach by reducing flooding, collision and traffic congestion. The work in this paper draws a bed on implementing a token based distributed approach to save energy at in order to mitigate flooding. Simulation studies of the proposed MAC protocol have been carried out using Castalia simulator.**

***Keywords* — WSN, MAC Protocol, Duty Cycle, Token, Traffic Congestion.**

I. INTRODUCTION

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. They have applications in a variety of fields such as environmental monitoring, military purposes and gathering sensing information in inhospitable locations. Sensor nodes have various energy and computational constraints because of their inexpensive nature and ad-hoc method of deployment.

Energy consumption is the most important factor to determine the life of a sensor network because usually sensor nodes are driven by battery and have very low energy resources. This makes energy optimization more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. Fairness is a critical issue when accessing a shared wireless channel. Fair scheduling must then be employed in WSNs to provide proper flow of information. A number of fair scheduling schemes exist in the literature; where some are centralized, and others are distributed. In general these fair scheduling schemes determine appropriate weights in order to meet QoS criteria. In most schemes weights are assigned and not updated for dynamic network conditions [1].

In WSNs, each sensor node can be in either of the mentioned states: Active (for receiving and transmission activities), Idle or Sleep. With the objective of prolong the life time of WSN, reducing the 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[5]. A node's main waste of energy is due to the following factors –

- Collision: Collision can occur when two nodes transmit data at the same time and interfere with each other's transmission.
- Idle Listening: A node wastes its energy when it is listening to an idle channel waiting for traffic.
- Overhearing: A node pines away its energy if it hears a packet which is not destined for it.
- Over-emitting: The energy can be wasted if a node sends data and the destination node is not ready to receive.
- Protocol Overhead: The contention-based protocols desolate energy because nodes use control packets (RTS, CTS, ACK) before sending the data.
- Traffic Fluctuation: The fluctuations of the traffic load can lead to the dissipation of a node's energy reserves. Therefore, the protocol should be traffic adaptive
- Flooding: While nodes send out the data packets at the same time to the sink that causes flooding resulting in to collision and therefore loss of data packets which leads the retransmission of the lost data packets from the nodes. The same ravages energy.

In this paper, we propose on energy efficient token based MAC protocol for WSNs, in order to reduce energy consumption of each sensor node which is one of the important issues to prolong the network lifetime. By allowing nodes to operate with a new token-based approach duty-cycle we intend to produce energy efficiency in an event based approach by reducing flooding, collision and traffic congestion. The work in this paper draws a bed on implementing a token based distributed approach to save energy at sink node in order to mitigate flooding at sink node. in this paper we premise token based approach in 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 can achieve better energy-throughput tradeoffs and extend node life span substantially while providing fewer collisions.

The work in this paper draws a bed on -

- Implementing a token based distributed approach to save energy at sink node in order to bring down flooding at sink node.
- The proposed approach also saves energy consumption rate of each node by curtility the number of data packet transmission.
- The effort is useful in environments where packet delivery is important with minimal congestion. Thus need to slim down probing message in the MAC layer implementation.
- By implementing token-based MAC, at once only a single source can interact with sink, thus cuts down congestion and ensures reliable data delivery in event based sensor networks where data sensed is more significant.

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

WSN consists of a large number of small and low-cost sensor nodes powered by small batteries and equipped with various sensing devices. Due to limited energy in WSN energy efficiency is an important factor in designing a MAC protocol. 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 all these above MAC protocols, due to the synchronized scheduling, transmission collisions, flooding will increase 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 [2], TMAC [2], RMAC [3] and HEMAC [4] 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 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
- Synchronization (Request and token allotment) and Data Transmission phase

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 and 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.

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.

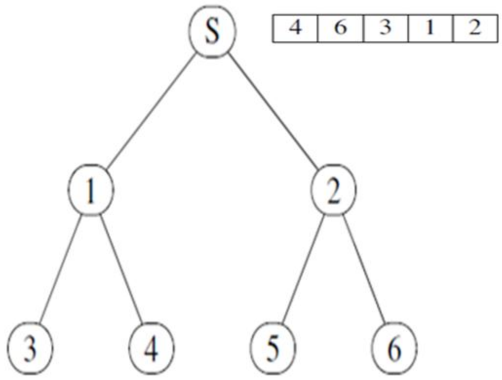

Fig1.Request packet enqueue at SINK node

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

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 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.

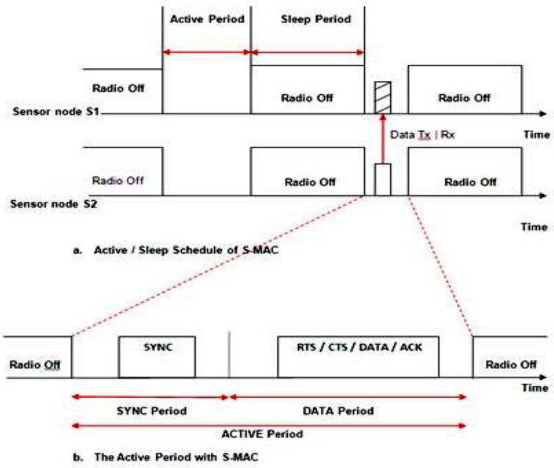

Fig.2: Active sleep scheduling in SMAC

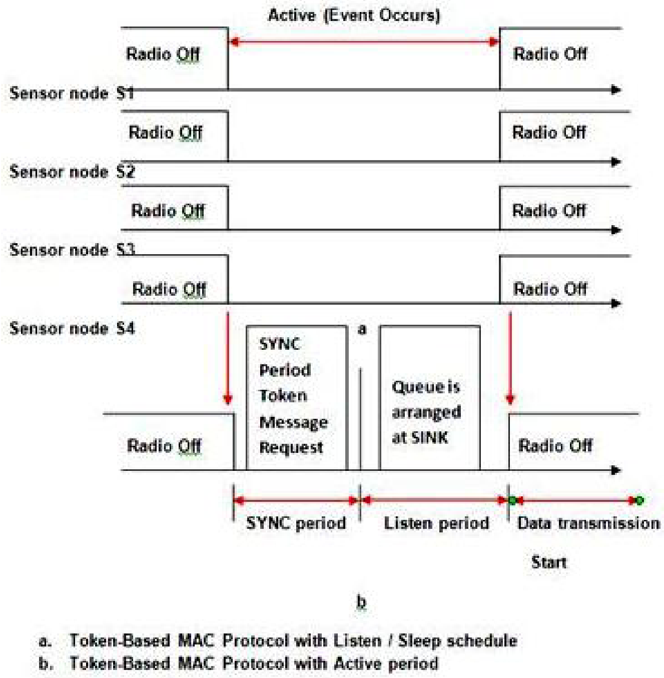

Fig.3: Listen sleep scheduling Token bases MAC

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.

1. *Flooding*

In case of many MAC protocols when an event takes place, at once all the sensor nodes start transmitting data packets to the sink resulting in Flooding at the sink and data packet loss at the sink node. On the contrary, in Token-Based MAC protocol we are employing synch RTS packet for requesting token from the sink. In this scenario, the node that holds the token can only transmit data packet. The outcome is at once only one node will be able to interact with the sink avoiding flooding. Flooding exhausts more energy as it demands the re-transmission of the data packets. Hence, non-occurrence of flooding automatically mitigates the energy consumption.

2. *Probing Message*

We require a minimum of 3 probing messages for a packet like RTS / CTS / ACK. In case of MAC protocol, we retransmit data packet when a packet loss takes place at sink and in turn the retransmitted data packet demands 3 more probing messages. On the contrary in case of Token-Based MAC protocol, retransmission of data packet is avoided as the data packet is transmitted only once resulting in collision to a lesser extent. Hence, in Token-Based MAC minimal probing message is used which results in preserving energy in the data communication.

3. *Duty Cycle*

In Token-Based MAC, the path and the node sequence is synchronized during the listen period. During this span only it is decided to which node the token need to be sent. The node which has got the token will transmit the data packet. When an event happens the nodes would wake up for a fixed time and get synchronized and the remaining time they will lie in sleep period. The data packet transmissions take place during the sleep period resulting in more sleep time with comparison to that of wake up / listen period. The same gives rise to mitigating energy consumption in this scenario.

4. *Idle Listening*

It does not happen here as the node which wants to transmit packet remains in sleep period. When it gets the token from the sink, it wakes up and carries out data transmission.

5. *Collision*

Collision is kept off here by employing the token mechanism. The nodes don't experience the same schedule for all and so they transmit data packets to the sink at different times which avoids collision. Here, the data transmission path could be known in advance employing synch RTS packet format. It results in forwarding the collision off at the sink node.

C. Proposed Algorithm

ALGORITHIM 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(HCj==∞)
            HCj=HCi+1;
            Pj=Ni;
            Broadcast ADVT(Nj,HCj);
        else if((Pj not sink) and (HCj>HCi+1))
            HCj=HCi+1;
            Pj=Ni;
            Broadcast ADVT(Nj,HCj);
        else
            Discard the ADVT;
        end if
    end if
```

Phase 2: Synchronisation and data phase

```
/* On receiving Request packet from node Ni to node
   Nj*/
if Request packet then
        if (Pi==Nj ) then
           Broadcast Request msg(RequiredNodeID ,Pj) ;
        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;
            If last data packet to send then
                TokenNode=false;
            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
```

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 [7]. 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 [8] data sheet and TelosB [9] 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

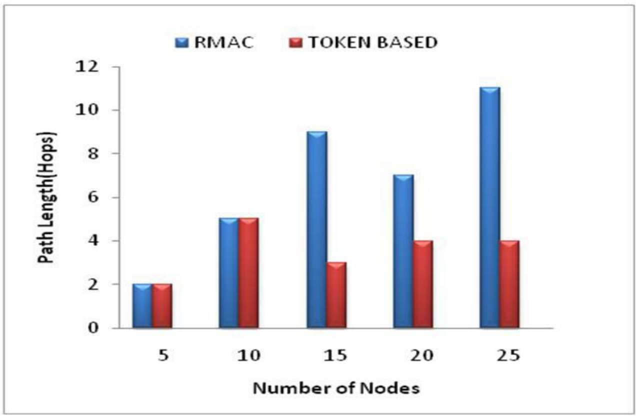

Fig 4: The Histogram of the path lengths of the realistic network

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.

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.

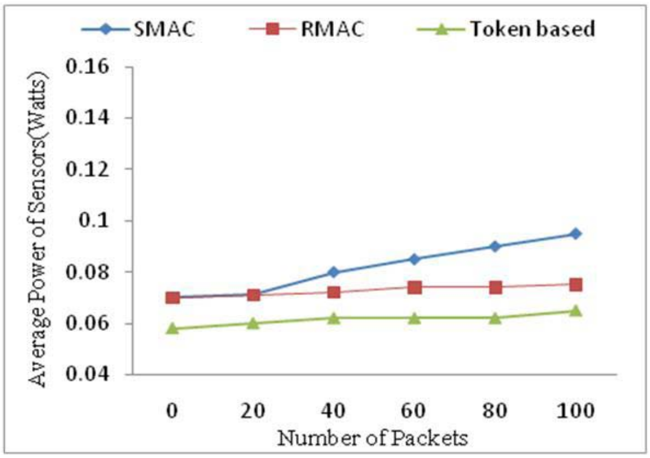

Fig 5: Average power of sensors

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.

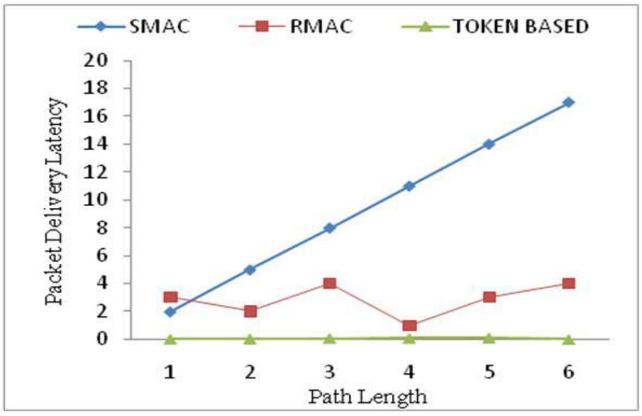

Fig 6: Delivery Latency

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.

D. Total energy consumption

In fig. 7 the total energy consumption of token based approach is directly proportional to the inter arrival time. After this also Token based MAC consume less energy than SMAC. Hence, this result happens not from the energy efficiency of SMAC but the inefficiency of SMAC in heavy traffic situations due to a large number of collisions, which will be clear in the below results.

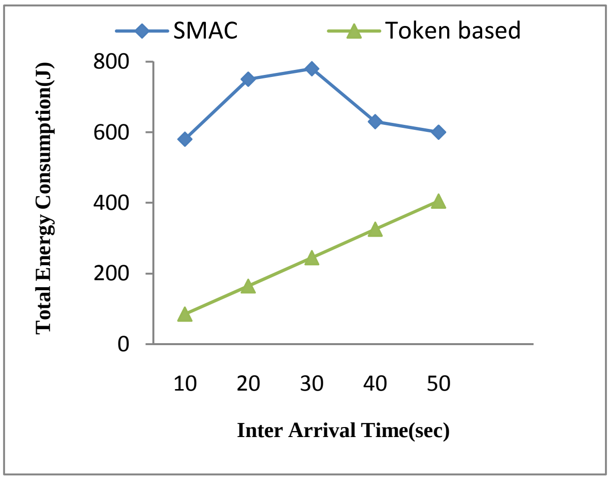

Fig. 7 Total Energy Consumption of SMAC and Token based MAC

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 are using a token to select a single sensor node which communicates with sink at a single time, which 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. We can also integrate sleep scheduling with token based MAC protocol to make it more energy efficient.

REFERENCES

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

[2]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.

[3] 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. [4]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.

[5] 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.

[6] 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.

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

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

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