PROPOSED MAC PROTOCOL FOR REDUCE ENERGY CONSUMPTION OVER WSN NETWORK

Ashutosh Gaur[1], Saroj Kumar[2]

[1]M Tech Student Computer Science Department, Babu Banarasi Das University, Lucknow U.P (India)
[2]Assistant Professor Babu Banarasi Das National Institute of Technology & Management, Lucknow U.P (India)

Abstract

Wireless sensor network consists of a large numbers of small electronically low cost sensor nodes which are sensing the environment physical measures with limited memory and limited processing unit. The efficient use of energy resources in sensor nodes could extend the lifetime of the WSN (Wireless Sensor Network). The main cause of energy waste is communication. Therefore, The issue of energy saving is considered primarily in transmission and dissemination of data in sensor networks. Using multi hop routing is always better, as it has many advantages over the single hop networks and all internal nodes bears the responsibility of data transmission. So, we proposed a protocol that provides reliable data transmission with higher data packet delivery rate with minimum consumption of energy. The proposed protocol uses multi token based MAC protocol that minimizes the flooding, collision and provides congestion less environment for data transmission. Each node has a unique ID, parents ID and a hop count number in network. Except the leaf nodes, Random sleep scheduling minimizes the idle listening and over hearing problem by putting some nodes into sleep mode. The main problem is Token distribution that leads to data loss and re-transmission, main causes of energy wastage. So, in this approach a token management is used that provides both reliable data transmission and higher delivery data packet ratio in WSN. Here, multiple parents of a sensor node support the fault tolerance and with token management, this proposed approach provides the reliable data transmission.

Keywords- *Energy Consumption; MAC (Media Access Control); Synchronized Scheduling; Token; Fault-Tolerant; WSN (Wireless Sensor Network); Fault Tolerant Multilevel Routing Protocol with Sleep Scheduling (FMS);*

--***--

1. INTRODUCTION

Wireless Sensor Network (WSN) consists of a large number of small and low cost sensor nodes which are equipped with various electronics devices that are able to interact with the surrounding environment by sensing or controlling physical parameters .These sensor nodes have to be work together for a particular application because a single node is incapable of doing so and they are uses wireless communication among them. A sensor node ensures the main functions like acquisition of environment physical measures, local processing of collected data and their transmission to the sink. Each sensor node carries specific tasks depending on different domain of specific applications such as monitoring physical or environmental conditions, sound, temperature, vibration and pressure etc. The data collected by each sensor node are reported to sink in a hop by hop fashion using wireless transmissions. Each sensor node consists of a processing unit, transceiver and a battery (2AA) for power supply. Processing unit of a sensor node has low powered unit so it supports low processing capability with limited memory like in a cell phones or in a Personal Digital Assistants (PDA). For communication among sensor nodes, they need a low energy supported medium. A sensor node is densely deployed in hostile environment and provides power communication by wired is not possible so they are supports wireless communication among them. As the size of each sensor node is small therefore battery size is also very small in nodes (Generally button or pencil cell) but provide the power through wired is not possible. Hence, such application must support energy efficient technique to maximize the sensor nodes and network life-time. Basically a WSN life time is depends on; when first node is dying or when the first link among the nodes is failed (disconnected). Transceiver (Radio) of a sensor node is the main power consuming unit even if transceiver is idle, so usually sensor nodes are put to sleep mode to conserve the energy when nodes are not transmitting/sensing the environment. Hence, to increase the life time of a wireless sensor network (WSN) we required a routing approach that supports the minimum number of re transmission and random sleep scheduling.

2. RELATED WORK

The routing protocol is supported the static nature of sensors node and application need the gather the information by sensor nodes when event is detected and information is transmitted immediately to the sink. Each node is associated with unique ID. Communications between nodes are symmetric and bidirectional and assumes clock is synchronized. Firstly, construction of broadcast tree after that transmission of data from source to sink whenever required. Tree is constructed in such a ways that internal nodes are in wakeup mode[3] and all leaf nodes are in sleep mode and energy level of internal nodes should be higher than leaf nodes. Tree is constructed in such a ways that exactly two parents of each node toward the sink, so system provides supports for fault-tolerance.

- Energy efficient single Token based MAC protocol integrates with sleep scheduling provides less energy consumption [7] so helps in prolonging of network lifetime and reduces the congestion in WSN in single low duty-cycle. Token based MAC protocol provides less congestion and main keys for its a Token (Request message) that used for synchronization of path and a Queue on sink that store requests from different sensor nodes. Each node associated with node ID, parent node ID and an array that stores data packets and reply message send through it.

Token based MAC protocol divided into 2 two phases:-

1. Level discover phase,
2. Synchronization with sleep scheduling and Data Transmission phase.

- The FMS protocol maintains network connectivity and provides reliable data transmission in event driven monitoring application WSN with a random sleep scheduling to provide energy efficiency. In sleep scheduling only radio is off but not sensing unit. FMS protocol basically divides into two parts:
  1. Level implementation and path establishment,
  2. Periodic sleep scheduling with data transmission.

Basically, FMS is working on level so level 1 node consume more energy than other because only one[8] hop distance between level 1 node and sink and traffic is very high in this region. If a node of level 1 is died then bottleneck problem is arise, to remove this problem protocol uses a random sleep scheduling with routing to overcome it.

- In single Token based MAC protocol, depending on token availability there are always an end to end communication between source sensor node and sink, still it would produce high latency. Multi-Token based MAC protocol, not only extend the network lifetime and maintain network connectivity but also ensure congestion less, fault tolerance and reliable data transmission. In this approach, we are taking some assumptions: the sensor nodes are densely and randomly deployed and each node is associated with unique node ID and for communication with neighbour mode is symmetric and bi-directional. Sink have more energy and powerful then all nodes for any tasks like processing communication capability, memory, power etc.[9] The multi token based MAC protocol consists four Different phases:
  1. Token management phase,
  2. Tree construction phase,
  3. Periodic sleep scheduling and data transmission phase,
  4. Multi Token based MAC protocol.

3. WORK PLAN

- Tree construction: Initially tree is constructed and re configured after a periods. This tree constructionuses BTC-1 and BTC-2 with level ordering to provide multiple path from a source node to sink.
- Token management: Multi-Token based protocol requires a better Token management that provides a reliable data transmission and increase the data delivery ratio.
- Random sleep scheduling: Random sleep scheduling provides some % of nodes which are put to sleep mode for a period and maintaining networks connectivity. It helps to prolong network life and ensure the balance energy consumption.
- Data routing packets: When a node detects an event and wants to transmit data packets to sink but the parent node is died then node can transmits data from its second parent node.
- Fault-tolerance: If the parent node (Pj) of a child node (Nj) is down then child node (Nj) can make a path from its neighbour node and transmits the data packets to sink.

4. PROPOSED WORK

- The proposed routing protocol is support the densely deployment of sensor nodes in event-driven WSN. The sensor nodes are uniformly randomly distributed in sensing area. The objective of proposed multi level routing protocol is to maintain connectivity of networks with minimum possible number of retransmission in WSN. Due to efficient token management, the proposed routing protocol can increased the data packet ratio .A random sleep scheduling is integrated with this protocol to economize the energy consumption during idle listening. Putting nodes to sleep will not prevent critical information to be sensed and forwarded to the sink since their sensing unit will be on and only the radio will be turned off.

We have to consider some assumptions for our objectives:

1) Sensor nodes are deployed in randomly and densely manner in WSN.
2) Each node should be with a unique node ID (Nj) and Level number (HCj).
3) Each sensor node has limited resources, while the sink nodes have powerful resources to perform any tasks and communicate with the sensor nodes.
4) Each node maintains two buffers, one for storing data packets and other for request message.

The proposed algorithms consist of three different phases:

1). Tree construction phase.
2). Token management phase
3) Random sleep scheduling and data transmission.

4.1 Tree Construction:

In this approach at first tree is constructed periodically, with minimum number of broadcast messages. We used the tree construction approach as in. The sink node considered as root node and each sensor nodes has two parents PFj,1 ,PFj,2 except sink with a level number (HCj) so that there are multiple paths exists from any node to sink. Each node assigned with a unique node ID (Nj) and a Level number (HCj) that based on number of hop count in path from sink to current sensor node. In beginning, set sink at Level 0 and parents ID is itself sink and all other sensor nodes has HC number is infinite and parents ID are -1 with cost field both as infinity. During the tree construction, when a sensor node selects its parent based on ADVT packet , its set own Level to one more than level assigned to its selected parents . By doing so, at the end of tree construction (using BTC-1 and BTC-2) each sensor node have a level number (HCj) and its parents (PFj,1, PFj,2). Once the tree is constructed, each intermediate node assigns with a Token and leaf node have to request for immediate parent sensor node for Token to start data transmission. For balancing the energy consumption of each sensor node, tree is re-constructed periodically and level number of each node after reconstruction may be changed except the sink. Since this MAC protocol represents the network into levels, the sensor nodes having level 1 may run out of energy before than other sensor nodes because these nodes have the responsibility of delivering data packets of all the sensor nodes in the network to sink. Thus, there are chances of creating bottleneck problem near the sink. Periodic random sleep scheduling scheme integrated with this protocol to reduce energy consumption of level 1 sensor nodes in the network and consequently address the bottleneck problem around the sink. This whole process models the network in different levels and implements a path from each sensor node to sink. If any intermediate node runs out of energy then the corresponding leaf nodes can establish a new edge which connects to another parent node at same level from the network and send data packet based on the availability of the token, so this approach is fault-tolerance and provide reliable data transmission to node failure.

```
Procedure BTC-phase1
begin
  CFj,1=CFj,2=1;
  if (First period) then
    PFj,1=PFj,2=1;
  end
  isActive = isBroadcasted = false;
  while (node j receive ADV1 (Ni, CFi,1, PFi,1) message from node i) do
    Set back off timer to ;
    if (isBroadcasted=false) then
      if (Ni is sink) then
        CFj,1=CFj,2=CFi,1+Cj;
        PFj,1=PFj,2=Ni;
        levelj = leveli+1;
      else if ((PFj,1 is sink and CFj,1 = 1) or (PFj,1 is not sink) ) then
        if ((CFi,1+Cj) < CFj,1) then
          CFj,1=CFj,2=CFi,1+Cj;
          PFj,1=PFj,2=Ni;
          levelj=leveli+1;
        end
      end
    else if ((Nj=PFi,1)) then
      isActive=true;
    end
  end
  if (back off timer expire and isBroadcasted=false) then
    isBroadcasted=true;
    Broadcast ADV1 (Nj, CFj,1, PFj,1) message;
  end
  if (Construction phase completed) then
    Break;
  end
  if (isActive) then
    Nj is an internal node;
  else
    Nj is a leaf node;
  end
end
```

Algorithm 1: Phase 1 of broadcast tree construction of node j

```
Procedure BTC-phase2
begin
  while (node j receives ADV2 (Ni,CFi,1) message from node i) do
    Set back off timer to + c;
    if ((PFj,1 = Ni) or (PFj,1 is sink)) then
      Discard ADV2 message;
    else if (Ni is sink) then
      CFj,1 = CFj,2 = CFi,1 + Cj ;
      PFj,1 = PFj,2 = Ni;
      levelj =leveli+1;
    else if ((PFj,1 = PFj,2) or ((PFj,1 PFj,2) and ((CFi,1 + Cj) < CFj,2))) then
      CFj,2 = CFi,1 + Cj ;
      PFj,2 = Ni;
      levelj =leveli+1;
    end
  end
  if (Nj is an internal node)) then
    Broadcast ADV2 (Nj ,CFj,1) message;
  end
  if (back off timer expire) then
    Break;
  end
  if (Nj is a leaf node) then
  end
end
```

Algorithm 2: Phase 2 of broadcast tree construction of node j

4.2 Token Management

After tree construction, sink at root (HCj = 0) and other nodes are divided into leaf and intermediate nodes. Except leaf nodes, every other nodes in the tree posses single token for reliable data transmission. Token of each intermediate node can be allotted to only one of the child node on the basis of FCFS (First Come First Serve), this token provides collisions free communication. For data transmission, each intermediate node situated between sources to sink

Node requires a token from its immediate parents (Pj). This process is continuing till all data packets are arrives at sink. It is also possible to integrate data at each intermediate node and send it to sink at once which also decreases data collisions in network. Token is granted for a fixed time period and when time period is expired child node have to return token to its immediate parent node so that token is allotted to next requested

Node with the help of Request queue, it avoids starvation. Four control messages are required for transferred token or data transmission between two nodes. REQUEST message is transmitted from child nodes (Nj) to parent for acquiring token. When parent (Pj) receives REQUEST messages from different child nodes then Token allotment is done on the basis of arrival time stamp through releasing a TOKEN message to node (Nj). After receiving a TOKEN message, child node (Nj) have to first sends a REPLY message to parent node then starts transmission of data packets through DATA messages. When the last data packet is transmitted by child sensor node to its immediate parent then TOKEN will attached to last data packet and transmit to parent sensor node.

```
/*When a node Ni received a Request from node Nj*/
begin
if (Pj==Ni ) then //if Ni is parent node of Nj
if (TokenPj=True) then
if(Request queue is empty)
Broadcast TokenPj (Nj, Pj ) packet;
TokenPj = false;
else if (allot the TokenPj to next max.Trequest time in the
request queue at
node Ni )
Broadcast TokenPj (Nj, Pj ) packet;
TokenPj = false;
end
else (Pj == sleep mode)
Put the Request in Request queue at Nj ;
end
end
end
```

Algorithm 3: When a node i received a REQUEST for Token from child node j

For reliable data transmission, we need to minimize the loss of control message (REQUEST, TOKEN, REPLY and DATA). If REQUEST message is lost or error packet received by parent (due to collision) then child node wait for some time until timer (TIMER 1) is expires and then again resend REQUEST packet to its parent. Token is granted on the basis of arrival time stamp. If Token is lost then again child nodes are send REQUEST for parents token after timer expired (TIMER 1). Otherwise if correctly received then node should broadcast a REPLY message to its parent that it acquire the token for Data transmission. But main problem raise when RELPY message is lost and this may be reason of lost of data because data transmission may start with sending REPLY message and child nodes assume its parent node receives it REPLY message correctly. So there may be four different situations can be possible when an REPLY message is lost:

(1) If child sensor node (Nj) received both tokens from its parents and keeps a token from its first parent node (PFj,1) and sends REPLY message that child node (Nj), acquires PFj,1 Token to both parent and start transmission of data packet with PFj,1. Child sensor node (Nj) transmits REPLY to PFj,2 that child node (Nj) accept Token from PFj,1 and discard PFj,2 token. After receiving REPLY, token of PFj,2 is now allotted token to next child node (Nk) according to REQUEST QUEUE at PFj,2.

(2) If child sensor node (Nj) received both parents token from its parents then keep a token from its first parent node (PFj,1) and broadcasts REPLY message that child node (Nj), acquires PFj,1 token and start transmission of data packet. Child sensor node (Nj) also send REPLY to PFj,2 that child sensor node (Nj) accept token from PFj,1 and discard PFj,2 token but REPLY message for PFj,2 is lost then PFj,2 regenerate token when TIMER 2 is expire and broadcast NEWTOKEN. When child node (Nj) received token with new sequence number then compare with currently acquire token if NEWTOKENseq.no.>TOKENseq.no. then child sensor node (Nj) again send REPLY message to PFj,2 that child sensor node (Nj) transmitted through Token of PFj,1. After receiving REPLY, Token of PFj,2 is now granted its TOKEN to next node according to REQUEST QUEUE at PFj,2.

(3) If child sensor node (Nj) receives both parents token and acquiring token of PFj,1 and send REPLY messages to both parent that child sensor node (Nj) acquires PFj,1 token but REPLY message for PFj,1 is lost and data transmission is start by child sensor node (Nj). After TIMER2 expire, node PFj,1 regenerate Token and broadcast and child sensor node (Nj) compare NEWTOKENseq.no. > TOKENseq.no. if condition is true then again child sensor node (Nj) send REPLY message to PFj,1 and retransmits all data packets again.

(4) If both the parents nodes of current requesting child node (Nj) are not received the REPLY message waits until TIMER2 is expires then regenerates the new token and broadcast it and when child sensor node (Nj) compare tokens sequence numbers, discard newly token and resend REPLY message of old token to both parents.

```
/*When a node Nj receives a Tokens from parent nodes
(PFj,1 or PFj,2) and Token-management*/
if (Token PFj,1==True Token PFj,2==True) then
if (Nj ! (TOKENPFj; 1 == True)) ==acquiretheT okenPFj; 1
BroadcastREPLY (Nj; TOKENPFj; 1) packet;
setT IMER2; ==T imerforREPLY
if (PFj; 1REPLY flag == TruePFj; 2REPLY flag == True)
setT IMER3; TOKENflag = False;
```

```
while (TIMER3!expirek there is more than on Data packet in Data Buffer)
do {
Broadcast DATA(PFj,1, TokenPFj,1, Data , Nj) packet ;
if (last DATA packet) then
stop TIMER 3 ; Nj(TOKENPFj,1ag= =True) ;
end
}
end
end
end
end

/*When TIMER 2 timeout*/
if (PFj,1 REPLYflag==True PFj,2 REPLY flag!=True) then
Broadcast TOKENPFj,2 (Nj, TOKENPFj,2 new seq no.) packet; //BRCT by PFj,2
if (TOKENPFj,1 seq no. < TOKENPFj,2 new seq no.) Then
Broadcast REPLY (Nj, TOKENPFj,1 seq. no.) packet ;
set TIMER 2;
end
else if (PFj,1 REPLY flag!=TruePFj,2 REPLY flag==True) then
Broadcast TOKENPFj,1 (Nj, TOKENPFj,1 new seq no.) packet; //BRCT by PFj,1
if (TOKENPFj,1 seq no. < TOKENPFj,1 new seq no.) then
Broadcast REPLY (Nj, TOKENPFj,1 seq no.) packet ;
set TIMER 2;
do{
Re-Broadcast DATA (PFj,1, TOKENPFj,1 seq no., DataTIMER3, N ) packet ;
} while (TIMER 3 expire or last packet of Data)
stop TIMER 3; TOKENPFj,1 flag=True ;
end
else if (PFj,1 REPLY flag!=True PFj,2 REPLY flag!=True)
Broadcast TOKENPFj,1 (Nj, TOKENPFj,1 new seq no.) packet;//BRCT by PFj,1
Broadcast TOKENPFj,2 (Nj, TOKENPFj,2 new seq no.) packet; //BRCT by PFj,2
if(TOKENPFj,1 seq no. < TOKENPFj,1 new seq no. or TOKENPFj,1 seq no.< TOKENPFj,2 new seq no.) then
Broadcast REPLY (Nj, TOKENPFj,1 seq no.) packet ;
set TIMER 2;
do {
Rebroadcast DATA (PFj,1, TOKENPFj,1 seq no., DataTIMER3, Nj) packets ;
} while (TIMER 3 expire or last packet of Data)
stop TIMER 3; TOKENPFj,1 flag=True;
end
end
end
```

Algorithm 4: Token management between node Nj and its parent node Pj When the remaining energy level of a node (Nj) is below the threshold level then token is allotted to node (Nj) which have least remaining energy level so that data can be collected before node (Nj) runs out of energy or die. When a node is going out of battery then to provide Fault-tolerance, node (PFj,1) must broadcast an ENERGYLOW message (PFj,1) to its child sensor node (Nj) to construct a new path from any other neighbor node (Nk). After that successive child node send a HELLO message (Nj) to its neighbor and wait for reply message i.e. NEIGHBOR message. NEIGHBOR message (Nk, HCk) from a neighbor node (Nk) contain its HCk and node ID (Nk) then node (Nj) make parent node to replied neighbor node (Nk). If level is not same then node (Nj) broadcast a LEVELCHANGE message otherwise no need to broadcast LEVELCHANGE message (Nj, HCj).

4.3 Random Sleep Scheduling and Data Transmission:

4.3.1 Sleep Scheduling:

When any event is detected at any sensor node then WSN should immediately transmitted the data from sensor nodes (Nj) to its parent node (Pj) and parent node (Pj) to its parent node until data is reaches up to sink node. Deployment of sensor nodes with level number (HCj) in densely WSN is complete then only we can start proposed sleep scheduling at sink. Proposed algorithms is executed at sink and produced each time a new sequence of random nodes number. This random sequence of different nodes number is used for putt them into sleep mode for avoiding idle listening, over hearing. In Multi-Token based multi-level MAC protocol working on level wise, hence level 1 node consume more energy than other because only one hop distance between level 1 node and sink and traffic is very high in this region. If a node of level 1 is died then bottleneck problem is arise, to remove this problem protocol uses a random sleep scheduling with routing to overcome it. To integrate sleep scheduling with this routing protocol, as tree construction is periodically so we need a mechanism that puts more number of randomly selected nodes into sleep mode. Before going to sleep mode, they have to send an ENERYLOW message to its child nodes so child node select the alternative path to transmit data packet to sink. Also if any active node runs out of energy then it should broadcast Energy flow to its child sensor nodes, hence connectivity is maintained. Thus it acts as a reliable and fault-tolerance to node failure. Through random sleep schedule we schedule 20% nodes into sleep mode that are select in randomly in a WSN. When an event occurs at the active source nodes the data is forwarded from source node to Parent node. If a node is put to sleep mode and an event occurs at that node it wakes up, transmit the packet and goes back to sleep till the next event occurs or next period starts whichever is early. For each tree reconstruction, the process to find out approximately 20% of total nodes which must be put to sleep mode .

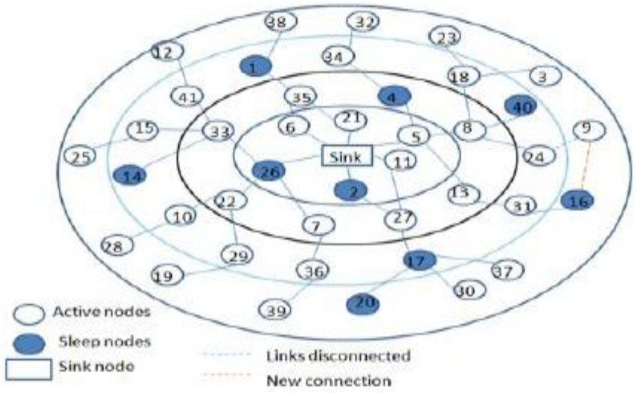

Fig.1. Proposed sleep scheduling example [3]

In Fig., if node number 24 is out of energy then node number 9 makes a path from node 16 by using a ENERGYLOW that transmit by node 24 to node 9 then node 9 sends HELLO message and when node 16 receives HELLO message it reply with NEIGHBOR messages and node 9 makes a path to node 16, this provides a fault-tolerance networks. Here level number is same so no need to broadcast a LEVELCHANGE message. It will increase the path length by 1 hop but provide reliable data communication and protect against data loss. We assume that approximately 20% of total nodes must be put to sleep. For this find the IDs of node those nodes are put to sleep through executing proposed sleep scheduling algorithms that produces the nodes sequence 17, 4, 40, 14, 1, 20, 2, 26 and 16. When a node in sleep mode and detects an event then simply put data in array and send request for Token to its immediate parents when Token is granted change its mode to active. In active mode nodes start transmission and return Token with last data packets. This random sleep scheduling 0algorithm generates a different sequence of nodes number at every execution time and then putts the selected nodes in to sleep mode. Hence, algorithms generate different sequence of node at each time so chances of network holes are very less as compared with previous sleep scheduling. Energy is conserved as more number of nodes are in sleep mode so network life time is increase by both factor i.e. nodes are live more and networks remain connected up to long time.

4.3.2 Data Transmission

When node (Nj) detects the event and wants to transmit data to sink. Firstly node (Nj) transmit requests for Token of its immediate parent PFj,1 or PFj,2 for acquiring Token or permission to send the data to its immediate parents and this process is repeated until data is reached up to sink. Node (Nj) can keeps Token of its parent for a fixed time period and after that node (Nj) have to return Token to its patent by attaching Token with last data packet. When a node (Nj) needs to transmits a data packet, following steps are executed:

(1) Node (Nj) sends a REQUEST message (REj; PFj,1 oPFj,2; Nj) to its parents node (PFj,1 or PFj,2) successfully.

(2)

(2) If REQUEST packet received by parent nodes PFj,1 or PFj,2 from node (Nj), it checks for the availability of its Token i.e. TokenPFj,1 or TokenPFj,2. If the Token is available at PFj,1, parent node PFj,1 sends the TokenPFj,1 using a TOKEN message (Nj, TokenPFj,1), else puts the request in its request buffer and waits for its Token to be available

(3) As soon as node (Nj) receives the TokenPFj,1 from its first parents, it sets a Timer3. After this, node (Nj) sends REPLY message to all immediate parents node that node (Nj) acquires the Token of PFj,1 and then starts sending Data message (PFj,1,TokenPFj,1, Data, Nj) from its data buffer to parent PFj,1. Token field in the parent remains false until Timer3 expires or node (Nj) has no data to transmits, whichever occurs first.

(4) But if REPLY message (Nj, TokenPFj,1) is lost and because data transmission from node (Nj) is continued, it causes the data loss. In this case, when Timer of REPLY Message is expires and those parents who didn’t get the REPLY message from node (Nj) regenerate the new Token with higher sequence number and resends. When node (Nj)receives NEWTOKEN compare its sequence number with previous. Node (Nj) resends REPLY message that node (Nj) acquires only PFj,1 Token and rest Tokens are discarded.After receiving REPLY, Token of PFj,2 is now allotted to next node (Nk) according to REQUEST QUEUE at PFj,2. When the node (Nj) sends its last packet to parent PFj,1,then it also returns TokenPFj,1 to its parent by setting the token flag to be true and attached the Token with last packet.

(5) On receiving the DATA packets, parent node (PFj,1) puts these packets in its data buffer. After getting the last data packet, parent node (PFj,1) also gets back its Token and provides the same Token to some other child node (Ni). If parent node PFj,1 is not a sink node, then parent node PFj,1 sends a request packet to its parents to forward or transmits the data packets that stored in its data buffer.

Repeat these steps up to data are reached received at sink.

5. CONCLUSIONS AND FUTURE WORK

Thus we have proposed an multi token based protocol with management and a routing protocol that incorporates random sleep scheduling to prolong the network lifetime. It maintains the connectivity of the network and the reliability of data transfer even when a node in the network runs out of energy. As discussed earlier, the major constraint in sensor networks is energy resource limitations which we overcome through this proposed protocol to possible extent by integrating a random sleep scheduling scheme and minimize the number of hops in path. We would like to focus on the following future works:

1. Working out the support of node with limited mobility.
2. To improve the random sleep scheduling algorithm with a view to include always a single connected components of networks

3. To improve the algorithm with a view to include the integration of data aggregation

NOMENCLATURE

Nj = jth sensor node
CFj,1 = Value of first cost field of node j
CFj,2 = Value of second cost field of node j
PFj,1 = Value of first parent node field of node j
PFj,2 = Value of second parent node field of node j
REj = Remaining energy of Nj
Cj =1/REj= Each nodes cost to be added to a path
Token Pj =Token availability at node Pj

REFERENCES

[1]. L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," ACM/Springer Mobile Networks and Applications, Vol. 11, No. 5, pp. 723-740, Oct. 2006.

[2]. E. Bulut and I. Korpeoglu, Dssp: A dynamic sleep scheduling protocol for prolonging the lifetime of wireless sensor networks, in Advanced Information Networking and Applications Workshops, 2007, AINAW07. 21st International Conference on, vol. 2,May 2007, pp. 725730.

[3]. Amulya Ratna Swain, R. C. Hansdah and Vinod Kumar Chouhan, An Energy AwareRouting Protocol with Sleep Scheduling for Wireless Sensor Networks in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on , April 2010, pp.933-940

[4]. J. C. Dagher, M. W. Marcellin, and M. A. Nei_eld, A theory for maximizing the lifetime of sensor networks," IEEE Trans. Commun., vol. 55, no. 2, pp. 323-332, Feb. 2007.

[5]. Akyildiz, I., Su, W., Sankarasubramaniam, Y. e Cayirci, E. A survey on sensor networks, IEEE Communications Magazine, pp. 102-114, 2002.

[6]. R. Shah and J. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in Proc. of IEEE WCNC02, Orlando, FL, March 2002, pp. 350355.

[7]. Soumya Ray, Subhasis Dash, Nachiketa Tarasia, Anuja Ajay, Amulya Ratan Swain, Congestion-less Energy Aware Token Based MAC Protocol Integrated with Sleep Scheduling for Wireless Sensor Networks, in World Congress on Engineering 2011 Vol. 2, WCE 2011,July 6-8, 2011, London, U.K., pp1755-1760,2011.

[8]. Anuja Ajay, Nachiketa Tarasia, Subhasis Dash, Soumya Ray and Amulya Ratan Swain, "Fault Tolera Multilevel Routing Protocol with Sleep Scheduling (FMS) for Wireless sensor Networks", in European Journal of Scientific Research ISSN 1450- 216X, Vol.55, No.1 (2011), pp.97-108.

[9]. Subhasis Dash, Amulya Ratan Swain and Anuja Ajay, Reliable Energy Aware Multi-Token based MAC for WSN, in IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), pp 144-151, 2012.