International Journal of Computer Applications (0975 – 8887)
Volume 33– No.4, November 2011

Cluster Based Task Scheduling in Wireless Sensor Network

Siddhartha Chauhan
Assistant Professor,
Department of Computer Science
and Engineering, and Engineering,
National Institute of Technology,
Hamirpur (H.P.), India.

Lalit Kumar Awasthi
Professor,
Department of Computer Science
and Engineering, and Engineering,
National Institute of Technology,
Hamirpur (H.P.), India.

ABSTRACT

Data Aggregation techniques are used in wireless sensor networks (WSNs) to collect the data samples from sensor nodes (SNs). Data aggregation techniques for WSNs must address to the issues of WSNs like limited energy, fast and efficient query response, which are essential for network's performance and maintenance. In this paper, we propose an energy-efficient cluster based task scheduling algorithm (CBTS) to reduce state transitions of cluster heads (CHs). Sensor nodes (SNs) consume different amount of energy in different radio states (transmitting, receiving, listening, sleeping and being idle).CBTS protocol reduces state transitions of radio, thereby reducing the energy consumption. CBTS schedules the CH activity so that when it is in wakeup state, all the reception and transmission activities are continuously performed before it goesinto sleep state. Wakeup time is reduced by CBTS Protocol. Simulation results show that CBTS reduces energy consumption and time delay for WSNs.

1. INTRODUCTION

The data flow in cluster based WSNs are from sensing nodes towards the sink through CHs. These CHs receive data from their cluster nodes (CNs) and other CHs. CH processes the received data and adds its own monitored data. This aggregated data is then sent by CH to its parent node. Scheduling of medium access plays an important role in the performance of WSNs. Time division multiple access (TDMA) and carrier sensing multiple access (CSMA) are two major medium access approaches in WSNs [1].

Major source of energy wastage in WSNs are packet collision, idle listening, over hearing, over emitting and re-transmission caused by co-channel interference. In packet collision, node receives more than one packet at the same time. All packets that collide have to be discarded and retransmission of these packets increases energy consumption. In ideal listening, nodes listen an ideal channel in order to receive possible data traffic. The third reason of energy wastage is overhearing [2]. Overhearing is caused when nodes receive packets which are not destined for them. The fourth reason of energy wastage is transmission of redundant data packets or redundant data to the destination. The fifth reason of energy wastage is over emitting [2]. Over emitting is caused by the transmission of data packets when the destination is not ready. The above factors of energy wastage should be considered while designing an energy efficient protocol for WSNs. If the network has limited number of non-overlapping channels then different channels cannot be assigned to every node. Retransmission of packets due to co-channel interference results in extra energy consumption [3].Medium Access Control (MAC) layer controls the radio of WSNs. In WSNs, during the data aggregation nodes listen to the channel every time even though data is not placed on the channel (ideal listening). TDMA scheduling removes the idle listening if WSNs are well synchronized.

Typically a SN wakes up periodically to sense the environment and generates a local sample stream ($\overrightarrow{X}$) at the rate of r samples per time period (T), to process the sensed data. The sensed data is then sent towards the sink by switching the radio in transmitting mode. Any node which has to receive the data has to switch the radio to receiving mode and process the data received. After finishing these operations, a SN will go in sleeping mode and again wakes up when it has to receive data or it has to sense any activity. Energy consumed by the nodes for waking up is more as compared to putting them in sleeping state. SNs wakeup for a small time but many times so, these state transitions consume considerable amount of energy.

In this paper our main focus is to reduce the state transitions in order to reduce energy consumption of CHs. CBTS reduces the state transitions as it schedules SNs in such a way that most of the nodes once awake finish all their desired operations.

2. RELATED WORK

The problem of efficient data gathering in WSN has been extensively investigated. Existing protocols can be divided into three categories [4]: (1) Cluster-based protocols, (2) Chain-based protocols, and (3) Tree-based protocols. Various scheduling methods for TDMA protocol with different objective have been proposed for WSNs. A staggered active/sleep scheduling has been proposed to solve the data forwarding interruption problem and enable continuous data forwarding on the multi-hop path in [5](DMAC). The principal aim of DMAC is to achieve very low latency for converge cast communications. In staggered active/sleep scheduling, RTS and CTS control packets are not used because they give unnecessary overhead but ACK packet and data retransmission are used to recover the lost data packet. DMAC achieves very good latency compared to other sleep/listen period assignment methods. DMAC does not utilize collision avoidance methods so, when nodes have similar schedule at same level in the tree, they try to transmit to the same node thus causing collision.

Traffic-Adaptive MAC protocol (TRAMA) [6] is a TDMA based algorithm proposed to increase the TDMA utilization in energy efficient manner. TRAMA uses a distributed election algorithm which is based on the traffic information to find the time slot for transmitting data within each two-hop neighbourhood. TRAMA ensures that all nodes in the one hop neighbourhood of the transmitter will receive data without any collision. Time is divided into random-access and scheduled-access (transmission) periods. Random access period refers signalling slots, which are used to obtain consistent two-hop topology information across all nodes. Transmission slots are used for collision-free data exchange and to schedule propagation.

Minimum Delay scheduling algorithm (MDS) [7] reduces the time delay in clustered wireless sensor network. MDS incorporates the slot reuse concept which significantly reduces the end-to-end latency. This protocol derives a relationship between the delay incurred by a data packet at each backbone network node (along its path to the sink) and the TDMA frame length. MDS reduces the time delay at the node and new data collected at the sink is very rapid.

A distributed aggregation scheduling algorithm (DAS) generates a collision free schedule for data aggregation in WSNs. The time latency of the aggregation schedule generated by this algorithm is minimized using a greedy strategy. DAS consists of two phases, one to construct a distributed aggregation tree and another to perform distributed aggregation scheduling. DAS has less time latencies and is collision free algorithm.

Energy Efficient Wakeup scheduling (EEWS) algorithm [9] reduces listening and wake-up time**.** When any node receives or transmits data then it has to be in wakeup mode. To reduce the energy cost, the activities of a subset of sensors are scheduled in one bundle, in which one node wakes up at most twice. The advantage of this protocol is that it is energy conserving. Energy is saved by reducing the state transitions which also dramatically reduces the cost of clock synchronization. This scheduling protocol also reduces time delay because slots are reused. Keshavarzian et al. [10] analysed different wake-up scheduling schemes and proposed a new scheduling method that can decrease the overall end-to-end delay. However, they did not consider the time-slot assignment problem for avoiding interference. TDMA-based wake-up scheduling can provide both energy-efficient and conflict free channel access [11], [12]. TDMA-based scheduling algorithms that minimize the number of time slots or the message delay are proved NP-complete [13], [14].

3. SYSTEM MODEL AND PROBLEM DESCRIPTION

The energy efficiency is a major issue for the protocol design for WSNs. The three main devices in wireless SN are radio, sensor and processor. These devices consume different amount of energy [15]. Sensor and processor energy consumption of a wireless SN are typically from the following operations: sampling the vicinity, reading sample data from the ADC, reading data from the flash, and writing/erasing data in the flash [16]. The energy consumption by the radio is from the following operations: transmitting a packet, receiving a packet and listening radio signals [17]. Since energy consumption by radio is much more as compared to other operations hence ourfocus is on energy consumption by the radio. The radio can be in one of the four states: transmitting, receiving, sleeping and listening. Every node consumes different amount of energy in these four radio states.

Assuming$E_{transmission}$, $E_{receive}$, $E_{listening}$, $E_{sleeping}$ be the respective energy consumed by the above stated four states of radio. Let$E_{sleep,receive}$, be the energy consumed by node for transition from sleep to receive state,$E_{sleep,transmit}$, and $E_{receive,transmit}$ be the respective energy consumed by node for transition from sleep to transmit state and receive to transmit state, whose typical values of the energy cost are summarized in table 1[18]. If the activity of every node is repeated after time period T then$T$ can be divided intot_s time slots. SN generates a local sample stream ($\overrightarrow{X_i}$) at the rate of $r$samples per time period T, with each sample a size of b bits. The maximum data rate supported by an RF transceiver for a SN is 40 kbps for Mica and Mica2, and 250 kbps for Micaz. Thus, the maximum data that can be transmitted in a time slot is about 150 bytes for Mica/Mica2 SNs and about 935 bytes for Micaz SN. The default data packet size by Tiny OS is 36 bytes [7], then under the ideal environment a node can transmit multiple data packets in one time slot. Typically when a node is transmitting packets to CH, some other neighbouring nodes that are in the listening state will also consume energy. Total energy cost by the node per time period $T(E_{i,T})$ is given by the following equation (1).

$$\begin{aligned} E_{i,T} = (E_{sense} * r * t_{sense}) + (E_{receive} * k * r * t_{receive}) \\ + (E_{process} * \big((k * r) + r\big) * t_{process}) \\ +(E_{transmission} * r * t_{transmission}) \\ +(n * E_{listening}) \quad (1) \end{aligned}$$

where k is number of nodes in a cluster, t_{sense} is the time taken to sense one sample,$t_{receive}$ is the time taken to receive one packet,$t_{process}$ is the time taken to process one packet, $t_{transmission}$ is time taken to transmit one packet and n is number of neighbour nodes. If all nodes are perfectly synchronized then there is no need for the nodes to be in listening state.

In TDMA, if node is well synchronized then radio will be in one of the three states: transmission, receiving, and sleeping. If node is transmitting data in the time slot t_s then we denote its state as $X_{i,trasmit,t_s} = 1$ otherwise$X_{i,trasmit,t_s} = 0$, if node is receiving data at time slot t_s then its state is denoted as $X_{i,receive,t_s} = 1$ otherwise$X_{i,receive,t_s} = 0$, likewise sleeping state of node during $t_s$is denoted its state as $X_{i,sleep,t_s} = 1$ otherwise$X_{i,sleep,t_s} = 0$. Node can change its states from sleeping to receiving, receiving to sleeping, sleeping to transmission, and transmission to sleeping. Hence at the time slot t_s radio can be in one of the three states: sleeping, receiving and transmitting and at the time slot $t_s + 1$ it can be in other state. The state transitions process consumes energy as summarized in table 1.

Table 1.Current consumption bynodes in various states and typical value of various symbols.

Symbol	Current consumption bySN	Symbol	Current consumption in SN	Symbol	Value
$E_{transmission}$	21.2 mA	$E_{sleeping}$	0.1 μA	t_b	416 μs
$E_{receive}$	13.3 mA	$E_{sleep,receive}$	13.3 mA	T	1s to 60s
$E_{process}$	2.1mA	$E_{sleep,transmit}$	21.2 mA	t_s	30ms
E_{sense}	2 mA	$E_{receive,transmit}$	21.2 mA	b	64 bits

$E_{listening}$	$5.8\mu A$	$E_{transmit\ ,receive}$	$13.3\ mA$	$packetLen$	36 Bytes

The energy consumed by the state transition from anyother state to sleeping state is very less as compared to other state transitions hence is neglected for the energy cost for state transition ($E_{transition}$) in time period$T$ given by equation 2.

$$E_{transiti\ on} = \sum_{t_s=1}^{T}(X_{i,sleeping\ ,t_s} \cdot X_{i,receive\ ,t_s+1} \cdot E_{sleep\ ,receive} + X_{i,sleeping\ ,t_s} \cdot X_{i,trasmit\ ,t_s+1} \cdot E_{sleep\ ,transmit} + X_{i,receive\ ,t_s} \cdot X_{i,trasmit\ ,t_s+1} \cdot E_{receive\ ,transmit} + X_{i,trasmit\ ,t_s} \cdot X_{i,receive\ ,t_s+1} \quad (2)$$

The main objective of this work is to minimize the energy cost (Equation 2) with fast and efficient query response.

4. CLUSTER BASED TASK SCHEDULING PROTOCOL

CBTS is an energy efficient protocol designed for scheduling the task of the nodes in order to reduce state transitions. CBTS initially arranges WSN as multilevel CH tree using BFS algorithm and then logically divides them into groups (Figure 1). CBTS algorithm reduces the wakeup time of nodes thereby reducing energy consumption of nodes, when the number of transmissions and receptions by a SN is fixed.

CBTS assigns levels to CHs and sink. Sink is at level 0. CHs one hop away from sink is at level 1 and two hops away are at level 2. Likewise all the CHs are assigned a level. The last level is represented by $maxLevel$ and current level of CH is indicated by$currentLevel$. The algorithm forCBTS is given in subsection 5.1. The$currentLevel$ is initially set at $maxLevel - 1$. All CHs at $currentLevel$ assign time slots to CHs at level $currentLevel + 1$. After scheduling these nodes, value of $currentLevel$ is set to $currentLevel - 1$ so all CHs at new $currentLevel$ schedule their descendent nodes, which are at level$currentLevel + 1$and$currentLevel + 2$.

Every CH (CH_j)at$currentLevel$ will calculate reception weight $T_{R,i}$(time slot required by CH at $currentLevel + 1$ for receiving packets from its CNs) and transmission weight $T_{X,i}$(time slot required by CH at $currentLevel + 1$for transmitting packets to it). CH_jcalculates reception time slot ($T_{R,i}$) and transmitting time slot ($T_{X,i}$) for everyCH (CH_i)at$currentLevel$ + 1. Total time slots T_i's required byeach CH at $currentLevel + 1$ is calculated as $T_i = T_{R,i} + T_{X,i}$. As shown in figure 1, CH_{12} is at $maxLevel$-1(where $maxLevel$ = 2)and CH_{22}, CH_{23}, and $CH_{24}$are at level 2. The $currentLevel$indicates the level of CH_{12}. The transmission rate for each node is same; hence parent of each node will be receiving in the same time slot in which its child node will be transmitting.CH_{12}, calculates reception weight ($T_{R,i}$) and transmission weight ($T_{X,i}$) of CH_{22}, CH_{23}, and CH_{24}. CH_{22}, has 5 cluster nodes so reception weight $T_{R,22}$ is 5. CH_{22} transmits all received and sensed packets in one time slot so $T_{X,22}$ is 1. Similarly, the reception and transmission weights for CH_{23}, and CH_{24}can be calculated as $T_{R,23} = 6$,$T_{R,24} = 6$, $T_{X,23}$=1and $T_{X,24} = 1$.

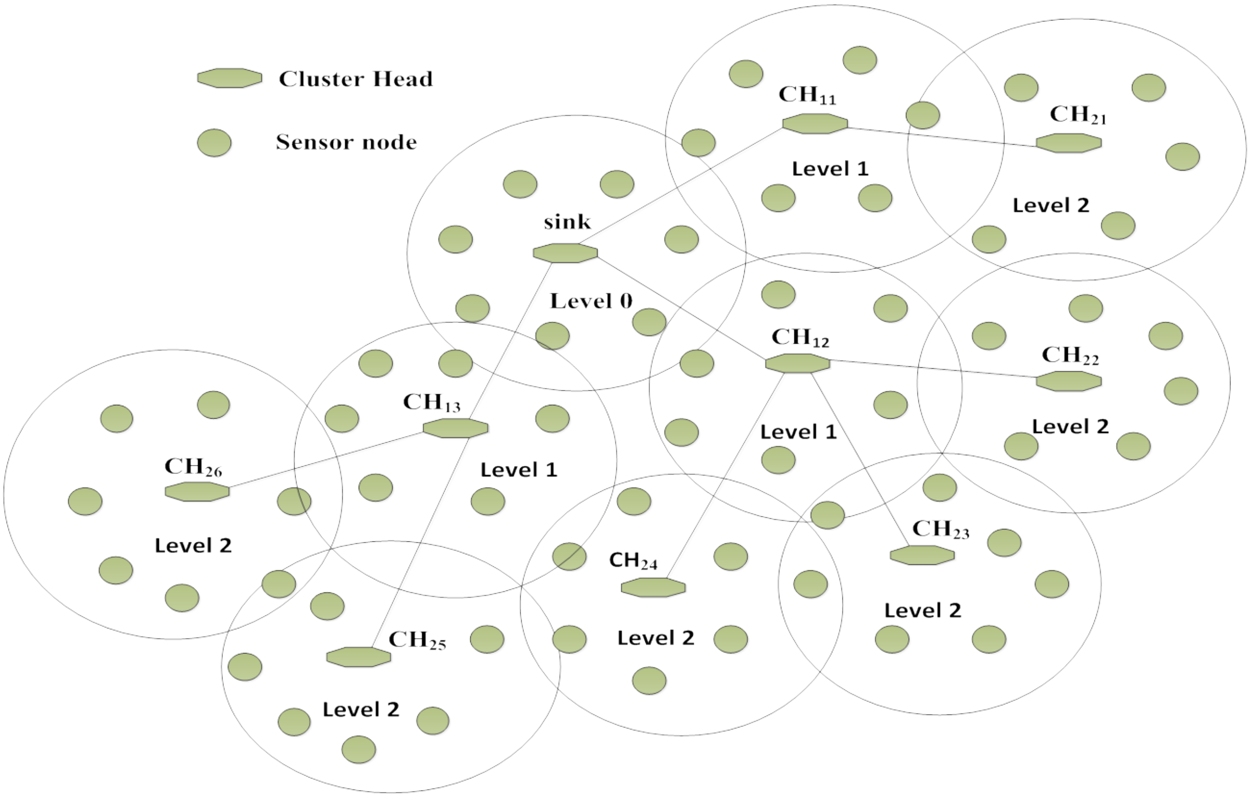

Fig 1:Multilevel tree with groups

The total time slots requirement of CH_{22}, CH_{23}, and CH_{24} are $T_{22} = 6, T_{23} = 7, T_{24} = 7$. The T_i's calculated by CH_{12} for its child nodes (T_{22}, T_{23} and T_{24}) are sorted in descending order (sort order). Thus the sort order is T_{23}, T_{24} and T_{22}. T_{23} and T_{24} have same time slot requirement so any one's T_i can be placed first in the sort order. Initially CH_{12} selects first child node i.e. CH_{23} from sort order and assigns 7 time slots to CH_{23} for reception as well as transmission (Figure 2). The first time slot is denoted by $T_{intial\,,i}$ and last time slot denoted by $T_{end\,,i}$ therefore $T_{intial\,,23} = 1$, and $T_{end\,,23} = 7$ for CH_{23}. Thereafter CH_{24} is assigned time slots as:

$$T_{intial\,,24} = T_{end\,,23} - T_{R,24} + 1 = 7 - 6 + 1 = 2, \text{ and}$$

$$T_{end\,,24} = T_{intial\,,24} + T_{24} - 1 = 2 + 7 - 1 = 8$$

Similarly CH_{22} is assigned time slots from $T_{intial\,,22}$ to $T_{end\,,22}$ as:

$$T_{intial\,,22} = T_{end\,,24} - T_{R,22} + 1 = 8 - 5 + 1 = 4, \text{ and}$$

$$T_{end\,,22} = T_{intial\,,22} + T_{22} - 1 = 4 + 6 - 1 = 9$$

CH_{22}, CH_{23}, and CH_{24} know their reception as well as transmission time slots allocated by CH_{12}, so they allocate their reception slots to their CNs. Figure 2 shows that CH_{23} has been allocated 7 slots. It will be allocating its reception slots (1-6) to its CNs for the reception of data from them. The slot 7 will be used by it for transmission of data packets to its parent CH_{12}. Similarly CH_{24} receives data packets continuously from its CNs in slots 2 to 7 and sends data packet in slot 8. CH_{22}, receives data packets from its CNs in slots 4 to 8 and transmits data packet in slot 9. CH_{12}, will be receiving data packets continuously from CH_{22}, CH_{23}, and CH_{24} in time slots 7, 8 and 9. All the CHs and sink will be scheduling their activity as explained above. Figure 2, shows that all the CHs at level 2 get continuous time slots to receive and transmit data packets. reduced. CHs at level 2 will have to wake up only once to receive and transmit data packets thus state transitions are effectively reduced.

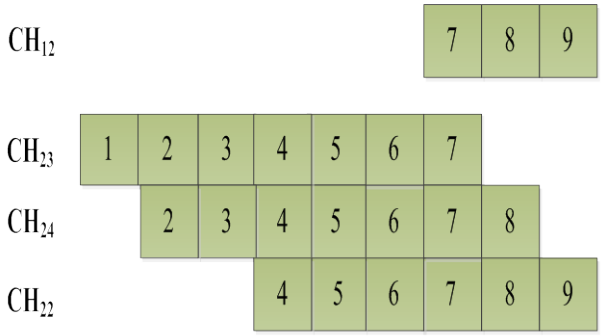

Fig 2.Time slots assignment to CHs for reception and transmission.

4.1 Algorithm for CBTS

```
1.  for $currentLevel = (maxLevel - 1)$ to 0 begin
    for $CH_p$ at $currentLevel$ begin
                    //$p$ =1 to $n$,No.of CHs at $currentLevel$
        for $CH_i$ at level $currentLevel + 1$ begin
                    //$i$=1 to $k$, No.of CHs at $currentLevel$ + 1
            Transmit $T_{R,i}$ and $T_{x,i}$ to $CH_j$
                    //$T_{R,i}$ is receiving time slots
                    //$T_{x,i}$ are transmitting slots
        end for
    end for
end for
2. $CH_p$ calculates $T_i = T_{R,i} + T_{X,i}$
        // $T_i$ is total time slot required.
3. $CH_p$ sort's value of $T_i$'s for CHs at $currentLevel$ + 1 in
    descending order as $T_1 > T_2 > . . . T_k$
                    //where $k$ = No. of CHs at $currentLevel$ + 1.
4. $CH_i$ with maximum $T_i$ from sort order gets time slot from
   $T_{initial,1}$ to $T_{end,1}$ for receiving and transmitting data.
                    //where $T_{initial,1}$= 1 and $T_{end,1}$= $T_1$
5.  for $CH_i$ from sort order where $i$= 2 to $k$. begin
                    // remaining CHs from 2 to k get
                    //time slot according to step 8.
        $CH_p$ assigns a range of time slot from $T_{intial\,,i}$ to
        $T_{end\,,i}$
        $T_{intial\,,i} = T_{end\,,i-1} - T_{R,i} + 1$
        $T_{end\,,i} = T_{intial\,,i} + T_i - 1$
                    // remaining $CH_i$ will receive and transmit
                    //data in time slot $T_{intial\,,i}$ to $T_{end\,,i}$.
end for
6. for $CH_i$ at level $(currentLevel + 1)$ begin
        $T_{X,CN} = T_{R,i} / m$
                    // $m$ is the number of CNs
                    // $T_{X,CN}$ is transmission slot of each CN of $CH_i$
                    //CNs of $CH_i$ will get transmission time slots
                    //according to $T_{R,i}$ of their CH.
end for
```

5. SIMULATION RESULTS

In state transition from sleep to wake up AT Mega 128L processor of Mica takes 4 millisecond and the RFM radio (used by Mica) takes 12 μs to switch between transmitting and receiving state. The raw bit time for RFM radio is 25μs. The Chipcon radio (Mica2 and Mica2 dot) takes 250 μs to switch between transmitting and receiving state and its raw bit time is 26 μs [9]. We conducted extensive simulations to compare the performance of CBTS with some methods earlier proposed. The maximum time delay and energy consumption for variable data rate and number of nodes was examined. Maximum time delay is the maximum time taken by any data packet to reach the sink. We have compared CBTS with node based and cluster based shortest path tree (SPT) algorithms [9].

5.1 Impact of data rate

The impact of data rate on time delay and energy consumption was examined for CBTS, node-SPT and cluster-SPT. The WSN was constructed by randomly placing 35 SNs in 5×5 square meter area. Simulation results were obtained keeping transmission range of the nodes as 1 meter. The results were obtained for variable data rates of links from 1 to 3.5 packets per slot. Figure 3 and 4 show the effect of variable data rate on time delay and energy consumption. Time delay and energy consumption decrease when data rate increases. Energy consumption and time delay by the node-based SPT is more because parent's wakeup time is more. CBTS algorithm has least time delay and consumes less energy due to small wakeup time and efficient slot reuse.

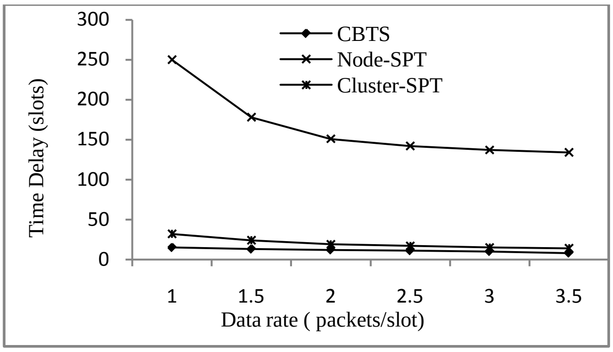

Fig 3: Impact of various data rates on the time delay.

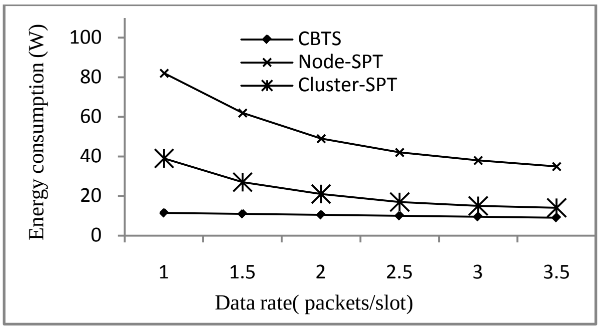

Fig 4: Impact of various data rate on the energy cost.

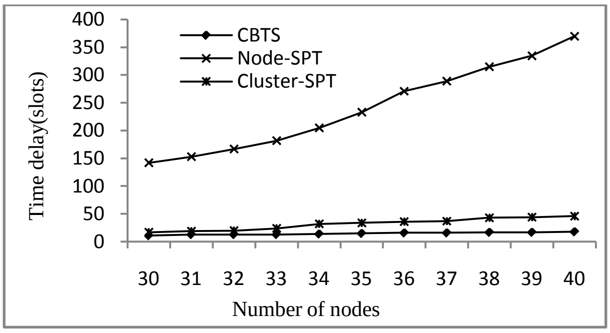

Fig 5: Impact of variable nodes on the time delay.

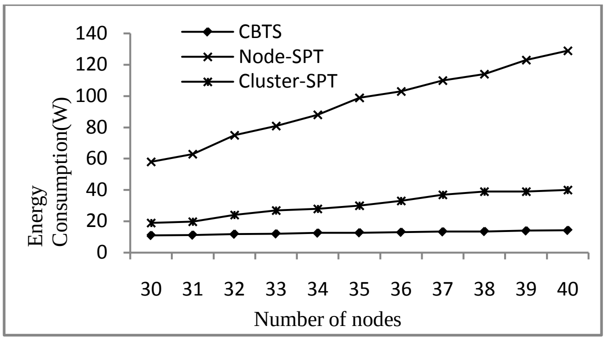

Fig 6: Impact of variable nodes on the energy cost.

5.2 Impact of numberof nodes

The impact of number of nodes on the energy consumption and time delay was examined through simulation. The number of nodes varied from 30 to 40 having transmission range of 1 meter. The results were obtained for data rate of link as 1 packet per slot. Figure 5 and 6 show that energy consumption and data delay increases when number of nodes increase. The traffic load in the network increases as the nodes increase. The time delay and energy consumption by the CBTS is less as compared to the other protocols. This is because CBTS requires less state transitions and all groups collect data independently.

6. CONCLUSION

In this paper we have proposed an energy efficient scheduling algorithm namely CBTS to reduce wakeup time of nodes by reducing the state transitions. CBTS initially arranges CHs using BFS algorithm into a multilevel CH tree. It then logically divides CHs into groups. WSN is then scheduled in order to reduce the state transitions. Simulation results show that our proposed algorithm reduces the energy consumption and time delay. CBTS is better in terms of energy consumption and time delay when compared with node-SPT and cluster-SPT algorithms. In future the focus can be on CBTS with reduced time latency and collision free data collection. The simulation results can be compared by doing experimental work.

7. REFERENCES

[1] Rhee, I., Warrier, A., Aia, M., Min, J.,and Sichitiu, M.L. 2008. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. IEEE/ACM Transactionson Networking.Vol. 16, No. 3. 511-524.

[2] Demirkol, I., Ersoy, C., and Alagoz, F. 2006. MAC Protocols for Wireless Sensor Networks: A Survey. IEEE Communications Magazine.115-121.

[3] Yen, H.H.,and Lin, C.L. 2009. Integrated channel assignment and data aggregation routing problem in wireless sensor networks. IEEE, In Special Issue on Wireless Ad-Hoc Networks.

[4] Khan, M., Pandurangan, G.,and Vullikanti, A. 2009. Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems. 20(1).124-139.

[5] Lu, G., Krishnamachari, B.,and Raghavendra, C.S. 2004. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks.In Proceedings of 18th International Parallel and Distributed Processing Symposium.

[6] Rajendran, V., Obraczka, K.,and Aceves, J.J.G.L. 2003. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks.In Proceedings of ACM SenSys ’03.181–92.

[7] Shi, L.,and Fapojuwo, A.O. 2009. TDMA Scheduling with Optimized Energy Efficiency and Minimum Delay in Clustered Wireless Sensor Networks. IEEE Transactions on Mobile Computing. Vol. 9, No. 7. 927-939.

[8] Yu, B., Li, J.,and Li, Y. 2009. Distributed Data Aggregation Scheduling in Wireless Sensor Networks.In

Proceedings of IEEE INFOCOM 2009. 2159-2161.

[9] Wu, Y., Li, X.Y., Liu, Y.H.,and Lou, W. 2010. Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation. IEEE Transaction on Parallel and Distributed System. Vol. 21, No. 2. 275-287.

[10] Keshavarzian, A., Lee, H., and Venkatraman, L. 2006. Wakeup Scheduling in Wireless Sensor Networks. In Proceedings of ACM MobiHoc. 322-333.

[11] Wang, W., Wang, Y., Li, X.Y., Song, W.Z., and Frieder, O. 2006. Efficient Interference Aware TDMA Link Scheduling for Static Wireless Mesh Networks. In Proceedings of ACM MobiCom.

[12] Djukic, P., and Valaee, S. 2007. Link Scheduling for Minimum Delay in Spatial Re-Use TDMA.In Proceedings of IEEE INFOCOM. 28-36.

[13] Arikan, E. 1984. Some Complexity Results about Packet Radio Networks. IEEE Transactions on Information Theory.Vol. IT-30, No. 4. 681- 685.

[14] Ephremedis, A., and Truong, T. 1990. Scheduling Broadcasts in Multihop Radio Networks. IEEE Transactions on Communications.Vol. 38, No. 4. 456-460.

[15] Sun, J.Z. 2009. QoS Compromise in Data Gathering for WSN, Mobile and Ubiquitous Systems: Networking & Services. In Proceedings of IEEE 6th Annual International conference MobiQuitous2009.1-2.

[16] Miller, M.J., and Vaidya, N.H., 2004. Efficient Bounds for the Stable Set Vertex Cover, and Set Packing Problem.In Proceedings of Wireless Communication and Networking Conference. Vol. 4. 2335-2340.

[17] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R.,and Anderson, J. 2002. Wireless Sensor Networks for Habitat Monitoring. In Proceedings of First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’02). 88-97.

[18] Bhattacharjee, D., Kumar, S., Kumar, A.,and Choudhury, S. 2010. Design and Development of Wireless Sensor Network. International Journal on Computer Science and Engineering.Vol. 02, No. 07. 2431-2438.

[19] Ergen, S.C.,and Varaiya, P. 2005. TDMA Scheduling Algorithms for Sensor Networks. Technical report. University of California, Berkley.