Optimization of State Transitions in Wireless Sensor Network

Sumit Kumar
Department of Computer Science and Engineering, National Institute of Technology, Hamirpur (H.P.), India
Email: sumittyagi999@gmail.com

Prateek Gupta
Department of Computer Science and Engineering, National Institute of Technology, Hamirpur (H.P.), India
Email: cseprateek@gmail.com

Siddhartha Chauhan
Department of Computer Science and Engineering, National Institute of Technology, Hamirpur (H.P.), India
Email: sidth@nitham.ac.in

***Abstract*--In wireless sensor networks (WSNs), data aggregation techniques are used to collect the data sample at sink through the cluster head, generated by sensor nodes. Data aggregation for WSNs must address to the issues of WSNs like limited energy in battery powered sensor, and fast and efficient query response which is essential to network performance and maintenance. In this paper, we propose an Energy Efficient Activity scheduling algorithm which is reduces the state transition of cluster head. During the data aggregation sensors nodes consume different energy in different radio states (transmitting, receiving, listening, sleeping, and being idle) and also consume energy for state transitions. Our proposed protocol Energy Efficient Activity scheduling (EEAS) protocol reduces state transitions of radio in cluster head(CH) there by reducing the energy consumption. Cluster head activity is scheduled so that in wakeup state reception and transmission activities are continuously performed by cluster head before going in to sleep state. Wakeup time is reduced by our proposed scheduling Protocol. Simulation results show that our proposed protocol reduces energy consumption and time delay.**

Keywords—Latency, State Transition, Collision, Medium Access, Scheduling.

I. INTRODUCTION

In cluster based WSNs, data flows from leaf nodes to sink node through the intermediate nodes are called cluster head. These cluster heads receive data from its child nodes and other cluster head. Cluster head process the data received from its child nodes and adds its own monitored data. This aggregated data is sent to its parent node. Scheduling of medium access plays an important role in the performance of wireless sensor networks. Time division multiple access (TDMA) and carrier sensing multiple access (CSMA) are two major medium access approaches in WSNs [1].

Major source of energy waste in WSNs are packet collision, idle listening, overhearing, overmitting, and retransmission caused by co-channel interference. In packet collision, node receives more than one packet at the same time then these packets collide with each other. All packets that cause the collision have to be discarded and retransmissions of these packets are required, which increases the energy consumption. In ideal listening, node listen an ideal channel in order to receive possible traffic. The third reason for energy waste is overhearing [2]. Overhearing occurs when a node receives packets which are going to other destination. The fourth reason of energy wastage is transmitted when redundant data packet or redundant data to the destination. The fifth reason for energy wastage is overemitting [2]. Overemitting is caused by the transmission of a data packet when the destination is not ready. The above factors of energy wastage do help to design an efficient protocol. If network has limited number of non overlapping channels than different channel is not assigned to every node on the data aggregation tree. Hence data aggregation introduces data retransmission which is caused by co-channel interference from neighbouring sensor nodes and extra battery power is required for retransmission of data [3].

The Medium Access Control (MAC) layer, which sits directly on top of the Physical layer and controls the radio of wireless sensor networks. In wireless sensor networks, during the data aggregation nodes listen the channel every time even if data is not placed on the channel is called ideal listening. TDMA scheduling removes the idle listening if networks are well synchronized.

Typically a sensor node s wakes up periodically to sense the environment (using the sensing device) and generates a local sample stream $\overrightarrow{X}$ at the rate of r samples per time period (T), to process the sensed data (using a computing component). The sensed data is then sent towards the sink node 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 data received by it from its child node. After finishing these operations, a sensor node will go in sleeping mode and again will wake up when data comes from another sensor node or it has to sense any activity. Protocols for WSNs should take into consideration the energy consumed in state transition (sleeping to wake up or wakeup to sleeping) of the nodes. Energy consumed by the nodes for waking up is more as compared to putting them in sleeping state. Sensor nodes wakeup for a small time but many times and these states transition from sleeping mode to wakeup mode consume more battery power. In this paper our main focus is to reduce the state transitions in order to reduce energy consumption of cluster heads. Our proposed protocol Energy Efficient activity Scheduling (EEAS) reduces the number of state transitions as it schedules sensor nodes in such a way that most of the cluster head once awake finishing all their operations. EEAS is efficient in handling of queries and its query response time is fast.

II. RELATED WORK

Recently, the problem of efficient data gathering in WSN has been extensively investigated. Existing protocols can be divided into three categories [2]: (1) Cluster-based protocols, (2) Chain-based protocols, and (3) Tree-based protocols. In this paper, our approach is schedule to the cluster based wireless sensor network. Various scheduling method for TDMA protocol with different objective have been proposed for wireless sensor networks such as DMAC[3], TRAMA[4], MDS[5], DAS[6], EEWS[7].

In DMAC[3] Protocol, a staggered active/sleep scheduling has been proposed to solve the data forwarding interruption problem thus enabling continuous data forwarding on the multihop path. Traffic-Adaptive MAC protocol (TRAMA)[4] 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 according to traffic information. The Minimum Delay scheduling algorithm (MDS)[5] is an energy efficient algorithm which reduces the time delay in clustered wireless sensor network. This scheduling algorithm incorporates the slot reuse concept which significantly reduces the end-to-end latency.

A distributed Algorithm generates a collision free schedule for data aggregation in wireless sensor networks. The time latency of the aggregation schedule generated by this algorithm is minimized using a greedy strategy. The distributed aggregation scheduling algorithm (DAS) [6], consists of two phases. First phase is to construct a distributed aggregation tree and second phase is to perform the distributed aggregation scheduling.

Energy efficient Wakeup scheduling (EEWS) algorithm [7] for sensor nodes is a TDMA protocol which reduces listening and wake-up time**.** To save energy consumption, the wake-up scheduling has been widely used. Keshavarzian et al. [8] analyzed different wake-up scheduling schemes and proposed a new scheduling method that can decrease the end-to-end overall delay. However, they did not consider the time-slot assignment problem to avoid interference. TDMA-based wake-up scheduling can provide both energy-efficient and conflict free channel access [9], [10]. TDMA-based scheduling algorithms that minimize the number of time slots or the message delay are proved NP-complete [11], [12].

III. SYSTEM MODEL AND PROBLEM DESCRIPTION

The energy efficiency is a major issue for the protocol design for WSNs. The three main devices in wireless sensor node are radio, sensor and processor. These devices consume different amount of energy [13]. Sensor and processor energy consumption of a wireless sensor node 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 [14]. The energy consumption by the radio is from the following operations: transmitting a packet, receiving a packet and listening radio signals [15]. We focus on energy consumption by the radio. The radio can be in one of the four states: transmitting, receiving, sleeping and listening and every state has different energy consumption (energy consumption per unit time). $E_{transmission}, E_{receive}, E_{listening}, E_{sleeping}$ be the energy consumed by the above stated four states of radio. Table 1 summarizes some typical values of the energy cost for different operations [16].

Let $E_{sleep,receive}$, be the energy consumed by node for transition from sleep to receive state. Like wise $E_{sleep,transmit}$, and $E_{receive,transmit}$ be the energy consumed by node for transition from sleep to Transmit state and receive to transmit state. Typically, the time to restart a sensor node from the sleep mode to active mode is about $4ms$.

If the activity of every node is repeated after time period T then T can be divided into time slots t_s. If node s_i 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 of a sensor node 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 sensor nodes and about 935 Bytes for Micaz sensor node. The default data packet size by TinyOS is 36 Bytes [7]. Then under the ideal environment a

TABLE 1. THE CURRENT CONSUMPTION FOR SENSOR NODE IN DIFFERENT STATE AND TYPICAL VALUE OF VARIOUS SYMBOLS.

Symbol	**Current consumption in sensor node**	**Symbol**	**Current consumption in sensor node**	**Symbol**	**Typical 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
$E_{process}$	2.1 mA	$E_{sleep,transmit}$	21.2 mA	t_s	30ms

E_{sense}	2 mA	$E_{receive,transmit}$	21.2 mA	b	64 bits
$E_{listening}$	5.8 μA	$E_{transmit,receive}$	13.3 mA	$packetLen$	36 Bytes

node can transmit multiple data packet in one time slot. Typically when a node s_i is transmitting packets to cluster head, some other neighbouring nodes that are in the listening state will also consume energy. Total energy cost by the node per time period T is $E_{i,T}$ is given by the following equation (1).

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

Where k is number of nodes in a cluster, t_{sense} is time taken to sense one sample, $t_{receive}$ is time taken to receive one packet, $t_{process}$ is 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 $X_{i,receive,t_s} = 0$, and if node is in sleeping state during t_s then we denote its state as $X_{i,sleeping,t_s} = 1$ otherwise $X_{i,sleeping,t_s} = 0$. Node can change its state: 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 some energy which is summarized in table 1. The energy consumed by the state transition from other state to sleeping state is very less as compare to $E_{sleep,receive}$, $E_{sleep,transmit}$, $E_{receive,transmit}$, and $E_{transmit,receive}$. Hence we can neglect the energy consumed by state transition from other to sleeping state. The energy cost for state transition by node s_i is $E_{transition}$ in time period T.

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

The main objective is to minimize the energy cost which is specified in equation 2. Another requirement is fast and efficient query response, so that every change in the environment can be detected immediately.

IV. ENERGY EFFICIENT ACTIVITY SCHEDULING PROTOCOL

We proposed a energy efficient protocol called Energy Efficient Activity scheduling (EEAS) protocol, which has been designed for scheduling the task of nodes. The Energy Efficient Activity scheduling algorithm reduces the state transition. This algorithm reduces the wakeup time of nodes thereby reducing energy consumption of nodes, when the number of transmissions and receptions by a sensor node is fixed.

Initially we arrange the cluster based wireless sensor network as Fig. 1. the In WSN, we assign the level to every cluster head and sink node at level 0. Cluster heads one hop away from sink is at level 1 and two hops away is at level 2 and so on. The last level is represented by the $maxLevel$ and

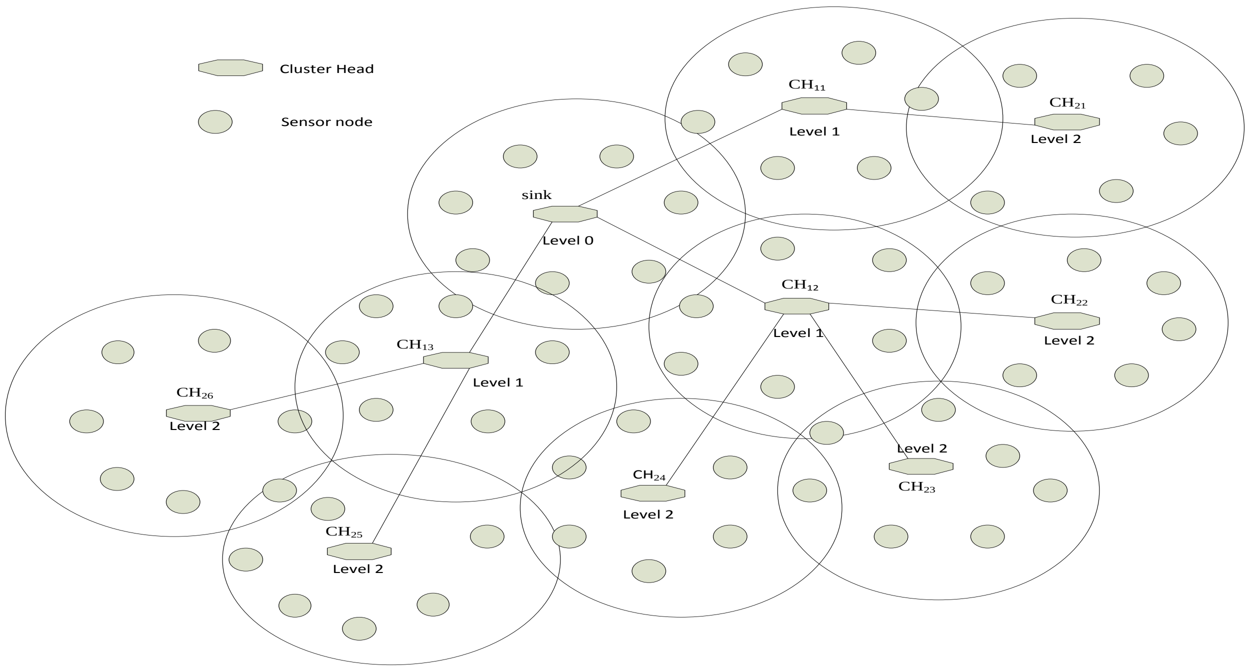

Figure 1. Cluster heads arranged as multilevel tree.

current level of cluster head is indicated by $currentLevel$. According to algorithm 1, Initially we set $currentLevel$ at the *maxlevel-1*. All cluster head at $currentLevel$ assign time slots to cluster head at they are level $currentLevel + 1$ to receive and transmit data packet. After scheduling these nodes, value of $currentLevel$ will be set to $currentLevel - 1$ then all cluster heads will be at new $currentLevel$ value and they will schedule their descendent nodes, which are at level $currentLevel + 1$ and $currentLevel + 2$.

Cluster head at level $currentLevel$ is represented by CH_j and cluster head at level $currentLevel + 1$ is represented by CH_i. Every cluster head CH_j at level $currentLevel$ will calculate reception weight $T_{R,i}$(time slot required by cluster head at level $currentLevel + 1$ for receiving packets from its child nodes) and transmission weight $T_{X,i}$ (time slot required by cluster head at level $currentLevel + 1$ for transmitting packets to it). Hence Cluster Head CH_jcalculates reception time slot $T_{R,i}$ and transmitting time slot $T_{X,i}$ for each cluster head CH_i which is level $currentLevel + 1$. Total time slot T_i for every cluster head CH_i can be calculated as $T_i = T_{R,i} + T_{X,i}$. After calculating T_i, Cluster head CH_j sorts T_i's of its child nodes in descending order. As shown in figure 1, cluster head CH_{12} which is at the $maxLevel - 1(maxLevel = 2)$ and cluster heads CH_{22}, CH_{23}, and CH_{24} will be at level 2. The $currentLevel$ indicates level of CH_{12} in clustered network. Since the transmission rate for each node is same and fixed, hence we assume that each node requires one time slot to transmit data at which parent node will also receive this transmitted data. Now CH_{12} can calculate reception weight $T_{R,i}$ and transmitting weight $T_{X,i}$ of CH_{22}, CH_{23}, and CH_{24}. The cluster head CH_{22} has 5 cluster nodes so reception weight $T_{R,22}$ is 5. It (CH_{22}) transmit the all receive packets and sense packet in one time slot so $T_{X,22}$ is 1. Similarly, we can calculate reception and transmission weight for CH_{23}, and CH_{24} as

$T_{R,23} = 6, T_{R,24} = 6$ and $T_{X,23}$=1, $T_{S,24} = 1$.

Algorithms 1: Pseudo code for Scheduling Clustered Network

1. For $(currentLevel = maxLevel - 1; currentLevel \geq 0; currentLevel - 1)$
2. For every Cluster head CH_j at level $currentLevel$(Where j repersent cluster head number at level $currentLevel$)
3. For each cluster head CH_i at level $currentLevel + 1$, CH_j calculates
4. Receiving slots $T_{R,i}$ and transmitting slots $T_{X,i}$ of n_i, where $T_{R,i}$= required number of time slots to receive data,

$T_{X,i}$= required number of time slots to send data

Total required time slot $T_i = T_{R,i} + T_{X,i}$

5. CH_j *sorts a*ll the value of T_i

$T_1 > T_2 > T_3 > T_4 > \ ....T_{k-1} > T_k$

Where k = number of cluster head at $currentLevel + 1$.

6. First node from sort order gets time slot $T_{intial,1}$ to $T_{end,1}$ time slot for receiving and transmitting data.

$T_{intial,1} = 1, T_{end,1} = T_1$

7. For each child node CH_i from sort order where *i= 2* to *k*.

8. CH_j assigns a range of time slot from $T_{intial,i}$ to $T_{end,i}$ to all CH_i as

$T_{intial,i} = T_{end,i-1} - T_{R,i} + 1$

$T_{end,i} = T_{intial,i} + T_i - 1$

CH_i will receive and transmit data from time slot $T_{intial,i}$ to $T_{end,i}$.

9. Nodes at level $(currentLevel + 2)$ will get time slot to send data according to as

For every cluster head CH_i at level $(currentLevel + 1)$

Receiving time slots of $CH_i = T_{R,i}$.

cluster head CH_i assigns these T_{Ri} time slots to its child nodes to transmit data.

Total required time slots T_i to every cluster head at level 3 can be calculated hence CH_{12} calculates it for cluster heads CH_{22}, CH_{23}, and CH_{24} as $T_{22} = 6, T_{23} = 7, T_{24} = 7$. CH_{12} after calculating total required time slots by its child nodes will be sort these T_i in decreasing order. The order will be T_{23}, T_{24} and T_{22}. Initially CH_{12} selects first cluster heads CH_{23} from sort order and assigns time slots to CH_{23} for reception and transmission (as shown in Fig. 2). The first time slot is denoted by $T_{intial,i}$ and last time slot denoted by $T_{end,i}$. Cluster head CH_{23} has 7 time slots to receive and transmit data packet. Hence $T_{intial,23} = 1$, and $T_{end,23} = 7$. CH_{24} will be assigned time slots from $T_{intial,24}$ to $T_{end,24}$ as

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

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

Then CH_{22} will be assigned time slots from $T_{intial,22}$ to $T_{end,22}$ as

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

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

Cluster heads CH_{22}, CH_{23}, and CH_{24} know their own reception time assigned by node CH_{12}, so they will assign same time slot to their child node to transmit data packet. The cluster head CH_{23} will receive data packet continuously from

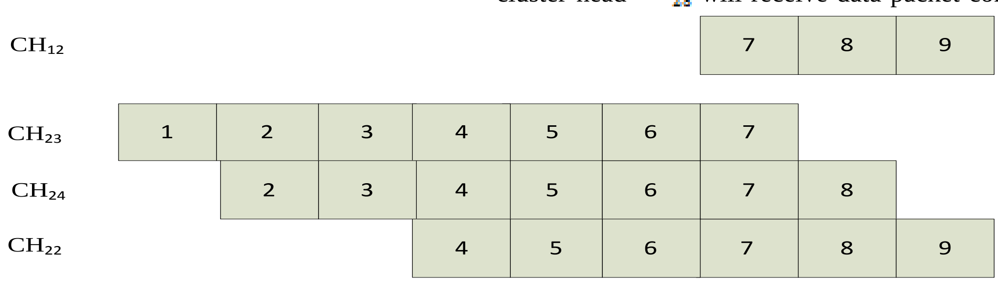

Figure 2. Time slots assignment to cluster heads for reception and transmission.

its child nodes at slots 1 to 6 and send data packet at slot 7 as shown in Fig. 2. Similarly CH_{24} receives data packets continuously from its child nodes in slots 2 to 7 and sends data packet at slot 8. The CH_{22} receives data packets from its child nodes at slots 4 to 8 and transmits data packet at slot 9. Hence, cluster head CH_{12}, will receive data continuously from cluster heads CH_{22}, CH_{23}, and CH_{24} in time slots 7, 8 and 9. Similarly entire cluster heads at level $currentLevel$ collects data from other cluster heads at level $currentLevel + 1$ and then variable $currentLevel$ will be set at level 0 and same process is used to aggregate data at this level.

As seen from Fig. 2 all the cluster heads at level 2 get continuous time slots to receive and transmit data packets. Thus state transitions have been effectively reduced. Cluster heads at level 2 will wake up only once to receive and transmit data packets. Hence this propose scheduling algorithm reduced the state transitions thereby reduces the energy consumption.

V. SIMULATION RESULT

In state transition from sleep to wakeup ATMega 128L processor of Mica processor takes of 4 millisecond and the RFM radio (used by Mica) takes 12 μs to switch between sending and receiving while the raw bit time is 25μs; and the Chipcon radio (used by newer Mica2 and Mica2 dot) takes 250 μs to switch between sending and receiving while the raw bit time is 26 μs[7]. EEAS protocol reduces state transition time to save the energy. We conduct extensive simulations to compare the performances of our proposed methods with some methods in the literature. In simulation result we examine the maximum time delay and energy consumption at various data rate and with various numbers of nodes. The maximum time delay is the maximum time taken by the data packet to reach the sink. We compare our multilevel tree based protocol with the node-based method provided in [17].

Impact of data rate: In this simulation we examine the impact of data rate on the time delay and energy consumption. We construct a network by randomly placing 35 sensor nodes in 5×5 square meter area. We set the transmission range 1 meter. We simulate sensor network when data rate of links varies from 1 to 3.5(packets/slot). Fig. 3 and 4 show that time delay and energy consumption with the data rate. The time delay and energy consumption are decreasing when data

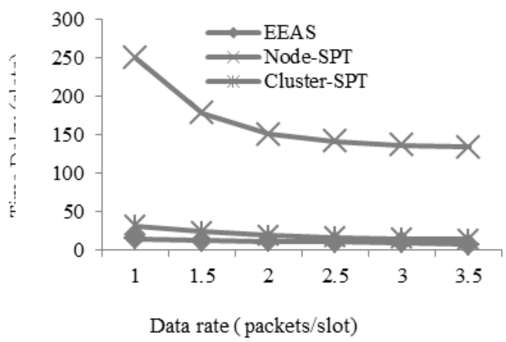

Figure 3. Impact of various data rates on the time delay.

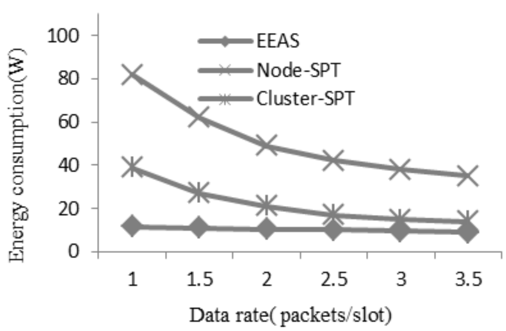

Figure 4. Impact of various data rate on the energy cost.

packet rate increasing. But reduction of energy consumption and time delay is slower than increasing data rate. Energy consumption and time delay by the node- based SPT is maximum because wakeup time of parent will be maximum. EEAS algorithm has smaller time delay and energy consumption due to small wakeup time and slot reusing.

Impact of Number of nodes: In this simulation we examine the impact of number of nodes on the energy consumption and time delay. We vary the Number of nodes from 30 to 40 in the sensor network. We set the transmission range 1 meter and data rates of links 1(packet/slot). In figure 5 and 6, we found that energy consumption and data delay increases when number of nodes increase. As the number of nodes increases traffic load in the wireless sensor network also increases. The Fig. 5 and Fig. 6 show that time delay and energy consumption by the EEAS is less as compared with the other protocols. This is because EEAS uses less state switches and all groups collect data independently.

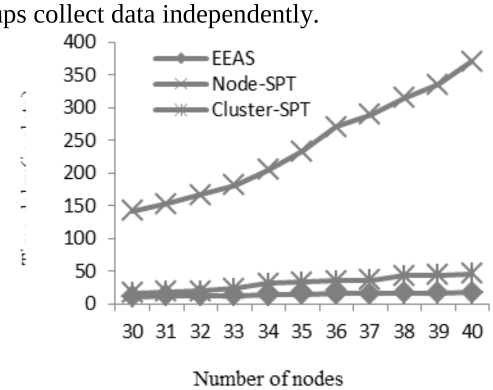

Figure. 5. Impact of various number of nodes on the time delay.

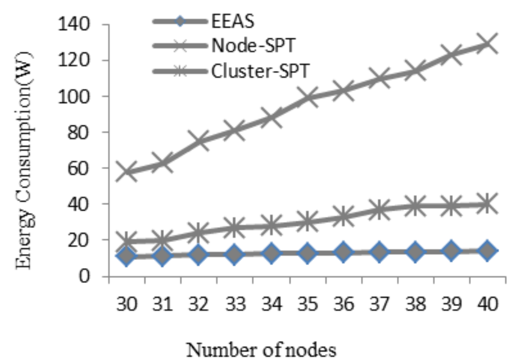

Figure. 6. Impact of various number of nodes on the energy cost.

VI. CONCLUSION

In this paper we propose a energy efficient algorithm which reduces the wakeup time of nodes by reducing the state transition. In this protocol, initially WSN is arranged as multilevel cluster heads tree using BFS algorithm and is then logically divide into the groups. WSN is then scheduled so that cluster heads nodes to reduce the state transitions. Simulation results show that our proposed algorithm reduces the energy consumption and time delay. EEAS is better in terms of energy consumption and time delay when compared with node-SPT and cluster-SPT based scheduling algorithms. In future we will be focussing on EEAS with reduced time latency and collision free data collection. The simulation results can be compared by doing experimental work.

References

[1] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," IEEE/ACM Trans. Networking, vol. 16, no. 3, pp. 511-524, June 2008.

[2] Maleq Khan, Gopal Pandurangan, Anil Vullikanti, Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Sensor Networks, IEEE Transactions on Parallel and Distributed Systems, 2009, 20(1):124-139.

[3] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks," Proc. 18th Int'l. Parallel and Distrib. Processing Symp., Apr. 2004, p. 224.

[4] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," Proc. ACM SenSys '03, Los Angeles, CA, Nov. 2003, pp. 181–92.

[5] Liqi Shi, and Abraham O. Fapojuwo, "TDMA Scheduling with Optimized Energy Efficiency and Minimum Delay in Clustered Wireless Sensor Networks", IEEE Transaction on Mobile Computing, Vol. 9, No. 7, pp. 927-939, July 2009.

[6] Bo Yu, Jianzhong Li, and Yingshu Li, "Distributed Data Aggregation Scheduling in Wireless Sensor Networks", in IEEE INFOCOM 2009, pp. 2159-2161, 2009

[7] Yanwei Wu, Xiang-Yang Li, YunHao Liu, and Wei Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," IEEE Transaction on Parallel and Distributed System, Vol. 21, No. 2, pp. 275-287, February 2010.

[8] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," Proc. ACM MobiHoc pp. 322-333, 2006.

[9] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient Interference Aware tdma Link Scheduling for Static Wireless Mesh Networks," Proc. ACM MobiCom, 2006.

[10] P. Djukic and S. Valaee, "Link Scheduling for Minimum Delay in Spatial Re-Use tdma," Proc. IEEE INFOCOM, pp. 28-36, May 2007.

[11] E. Arikan, "Some Complexity Results about Packet Radio Networks," IEEE Trans. Information Theory, vol. IT-30, no. 4, pp. 681- 685, July 1984.

[12] A. Ephremedis and T. Truong, "Scheduling Broadcasts in Multihop Radio Networks," IEEE Trans. Comm., vol. 38, no. 4, pp. 456- 460, Apr. 1990.

[13] Jun-Zhao Sun, "QoS Compromise in Data Gathering for WSN," Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International, IEEE 2009, pp. 1-2.

[14] M.J. Miller and N.H. Vaidya, "Efficient Bounds for the Stable Set, Vertex Cover, and Set Packing Problem," Proc. Wireless Comm. and Networking Conf., vol. 4, pp. 2335-2340, 2004.

[15] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02), pp. 88-97, 2002.

[16] Dipanjan Bhattacharjee, Sourabh Kumar, Akash Kumar, Sushabhan Choudhury, "Design and Development of Wireless Sensor Node," International Journal on Computer Science and Engineering, Vol. 02, No. 07, 2010, pp. 2431-2438.

[17] S.C. Ergen and P. Varaiya, "TDMA Scheduling Algorithms for Sensor Networks," technical report, Univ. of California, Berkley, 2005.