Modified Centralized Set Cover based Approximation(CSCA) for Duty-Cycled Wireless Sensor Networks

Rifatul Islam, Umama Tasnim, Mosarrat Jahan and Shaily Kabir
Department of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh
Email: shovonis@yahoo.com, mousumitasnim10@gmail.com, mosarrat@cse.univdhaka.edu, shailykabir2000@yahoo.com

Abstract— **Broadcast operation is one of the most fundamental operations in wireless sensor networks for data communication. In this paper we study the problem of finding an energy-efficient broadcast algorithm for duty-cycled wireless sensor networks. In such networks, sensor nodes usually stay in low energy sleep mode and occasionally wake up for a short period of time to sense, transmit or receive data. We investigate for broadcast algorithms that will reduce the number of data transmissions in such networks and thus decrease energy expenditure further. Finding such an algorithm is difficult because a node can either transmit or receive during its active time slot and moreover all of its children may not be active at the same time period. A node may need to wake up several times to transmit a single packet to all its neighbors which is very energy expensive. In this paper, we proposed a polynomial time algorithm known as Centralized Set Cover based Approximation with Minimum number of transmissions (CSCA_MT). Our algorithm actually modifies the existing CSCA [1] algorithm to reduce the number of transmissions to a greater extent. We conducted extensive simulations to evaluate the performance of our proposed algorithm and compared the performance with that of CSCA algorithm. Analysis of the result shows that CSCA_MT algorithm outperforms CSCA in terms of number of transmissions while being competitive in terms of average height of the broadcast tree and average delay.**

Keywords-duty-cycled wireless sensor network, data transmissions, broadcast, delay.

I. INTRODUCTION

A key concern for long operational lifetime of wireless sensor networks (WSN) is to utilize the limited battery power of a sensor node in sensible ways. As WSNs are often exploited in difficult to access regions it is not always possible to replace or recharge battery and thus wise utilization of energy is essential. Energy can be conserved into a number of ways in WSNs. A well-known scheme is to *duty cycle* the WSN. In a duty-cycled WSN, every node is accompanied with a *Wakeup Schedule* of fixed length. A node operates by repeatedly executing this wakeup schedule during its whole lifetime. Usually a node is kept in sleep mode by switching its internal circuitry off except a timer to wake it up. A node wakes up during its predetermined active time slot. During this time period, a node can sense data, transmit or receive data. A node can transmit data at any time but can receive data only during its predetermined active time slot [2]. Energy consumption in sleep mode is much less as compared to that of active mode. The amount of time a node will remain active depends on the data requirements of the applications. The percentage of time a node remain active during its whole life cycle is known as duty cycle [2].

Broadcast is an elementary operation of WSNs in order to establish communications such as to propagate interest and query messages for data acquisition, to pass control messages during network configuration time and to synchronize nodes to monitor certain events [2]. In this paper, we explore the problem of energy-efficient broadcast operation in duty-cycled WSNs by constructing a broadcast tree with the aim of reducing number of data transmissions. We formalize the problem in the next section.

A. Problem Formulation

A *Duty-Cycled WSN* is represented by a unit disk graph $G = (V, E)$ where V is the set of nodes in the network and $E \subseteq V \times V$ is the set of links. Each node $u \in V$ is associated with a *Wakeup Schedule* M_u of length n where, $M_u[i] = 1$ when u is active in time slot i and $M_u[i] = 0$ when u is in sleep mode. Node u operates throughout its lifetime by following M_u in a cyclic fashion. Our purpose is to provide broadcast service with minimum energy consumption in duty-cycled WSNs. Usually a *Broadcast Tree* is used as a backbone structure for performing broadcast operation in WSNs. A broadcast tree is a directed spanning tree rooted at a source node such that a packet from the source node can be sent to every other node in the network through the edges of the tree. As efficient utilization of energy is a major concern for long network lifetime of WSNs, it is also essential to reduce energy consumption for broadcast operation, too. One way to save energy is to reduce the number of data transmissions. We define *Cost (T)*, the cost of a broadcast tree T as the total number of data transmissions needed to complete the broadcast operation. Thus the problem we address is: Given a duty-cycled network $G = (V, E)$ where every node u is accompanied with a wakeup schedule M_u of length n and a source node $s \in V$, find a broadcast tree T rooted at s for G such that $Cost(T) \le Cost(T')$ for all broadcast trees T' for G.

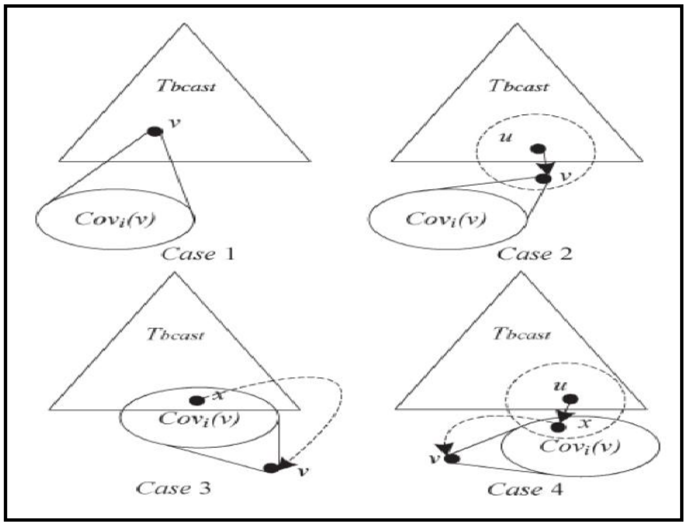

Figure 1. Four Cases to Connect $Cov_i(v)$ to the Existing Broadcast Tree T_{bcast} through v in CSCA Algorithm

TABLE I. NOTATIONS USED FOR CSCA_MT

V	Set of nodes
N	Total number of nodes
T	Length of wakeup schedule
$T_a(u)$	Set of active time slots of the adjacent nodes of u
U_i	Set of nodes with active time slot i
C_i	Set of covering nodes for U_i
$Cov_i(u)$	Set of nodes covered by u at time slot i

B. Our Contributions

In this paper, we investigate the problem of finding energy-efficient solution for broadcast problem in duty-cycled WSNs. Hong et el. proved that finding broadcast schedule with minimum number of data transmissions is a NP-Complete problem [1]. We propose a polynomial time algorithm: Centralized Set Cover based Approximation with Minimum number of Transmissions (CSCA_MT). This algorithm is actually a modified version of the existing Centralized Set Cover based Approximation (CSCA) algorithm [1]. Our experimental results exhibit that CSCA_MT reduces the number of data transmissions significantly as compared to the CSCA algorithm. Performance of CSCA_MT is competitive in terms of average delay and average height of broadcast tree.

The remainder of this paper is organized as follows. In Section II we discuss related works and in Section III we present CSCA_MT algorithm in detail. The simulation results are explained in Section IV. Finally we end the paper in section V with some concluding remarks and future directions for research.

II. RELATED WORK

Broadcasting in duty-cycled WSNs is an interesting problem. Han et el. studied both one-to-many and all-to-all multicasting problems for duty-cycled WSNs in [3]. The authors formalized one-to-many multicasting problem as Minimum-Energy Multicast Tree Construction and Scheduling (MEMTCS) problem and proposed a polynomial time approximation algorithm with a performance ratio of $O(H(\Delta + 1))$, where $H(\cdot)$ is a harmonic function and Δ is the maximum node degree of the network. For all-to-all multicasting, the authors presented an approximation algorithm that has the same approximation ratio as that of the proposed algorithm for the MEMTCS problem. They provided the distributed implementation of both algorithms and showed through experimental results that the proposed algorithms significantly reduce total transmission energy with good delay performance. The authors of [4] investigated the Sleeping Schedule-Aware Minimum Latency Broadcast (MLB-SA) problem in WANETs and proved that MLB-SA is a NP-Complete problem. They proposed Simple Layered Coloring Algorithm (SLAC) and Enhanced Layered Coloring Algorithm (ELAC) to address MLB-SA problem. Duan et el. [5] addressed Minimum Latency Broadcast Scheduling in Duty-Cycle (MLBSDC) multi-hop wireless networks. The authors proposed a generalized framework for MLBSDC problem that reduces delay as well as number of transmissions. In [6], the authors proposed an algorithm for broadcast operation in duty-cycled WSNs by considering the problem as a shortest path problem in a time-coverage graph with the aim of minimizing delay and forwarding cost. Research are also conducted to design energy-efficient MAC protocols for duty-cycled WSNs. Ye et el. [7] proposed S-MAC protocol that is designed for duty-cycled WSNs with the purpose of reducing power consumption for sensor nodes. The authors put emphasis on saving energy rather than the access fairness of nodes and broadcast latency that are considered as the most important performance criteria of MAC protocols for other wired and wireless networks. They identified the major sources of energy loss and designed a number of features to mitigate their impacts.

We compare the results of our algorithm with existing Centralized Set Cover based Approximation (CSCA) algorithm [1] in literature. Hong et el. considered the problem of constructing a broadcast tree in a uncoordinated duty-cycled wireless ad hoc or sensor networks that will minimize the number of node transmissions. The CSCA algorithm works in two phases. Set U_i represents a set of nodes in V that are active at time slot i and $Cov_i(v)$ is the set of nodes with active time unit i that are within the transmission range of node v. In phase 1, CSCA determines for each U_i a minimum covering node set C_i in a greedy fashion such that $\bigcup_{v \epsilon C_i} Cov_i(v) = U_i$. In phase 2, a broadcast tree is constructed by connecting all $\bigcup_{i \epsilon T} C_i$ to a source node s using the four cases shown in Fig. 1. Here T is the length of wakeup schedule.

Algorithm 1 Construction of Minimum Cover Set C_i

Input: A duty cycled WSN $G=(V, E)$ and U_i, $1 \leq i \leq T$

Output: Cover sets C_i, $1 \leq i \leq T$

Step 1) $C_i \rightarrow \emptyset$, $J \rightarrow V$ and $I \rightarrow U_i$

Step 2) Select a node $v = \arg max_{v \epsilon J} |Cov_i(v)|$

Step 3) $C_i \rightarrow C_i \cup \{v\}$, $J \rightarrow J \setminus \{v\}$, $I \rightarrow I \setminus Cov_i(v)$

Step 4) If $I = \emptyset$ then

Stop and output C_i.

Else

Go to Step 2.

Algorithm 1. Pseudo code for Constructing Cover Set C_i

III. ALGORITHM

In this section we present the polynomial time heuristic algorithm. Some basic notations used to describe the algorithm are presented in Table I.

A. Centralized Set Cover Based Approximation with Minimum Number of Transmissions (CSCA_MT)

This algorithm is analogous to CSCA [1] algorithm and differs in the way nodes are connected to broadcast tree T_{bcast}. Similar to CSCA algorithm, minimum covering node set C_i for each U_i, $1 \leq i \leq T$ is constructed using the algorithm described in [1]. Pseudo code for constructing minimum size cover sets is described in Algorithm 1. The aim of the algorithm is to cover all nodes $u \epsilon U_i$ with minimum number of data transmissions. Algorithm 1 always includes a node v in C_i that covers maximum number of nodes in time slot i with a single transmission. After constructing C_i, $1 \leq i \leq T$, T_{bcast} is generated by assigning priorities among the nodes. A stack S is used to keep track of the nodes to be added into T_{bcast}. A node v is pushed into S when it is considered to be added to T_{bcast} and popped out when its broadcast operation is finished. Initially, source node s is pushed into S. CSCA_MT repeatedly pops a node v from S and processes it using Step 1 described below:

Step 1:

- [Case 1] If a node $v \epsilon T_{bcast}$ and $v \epsilon C_i$, then connect $v \rightarrow Cov_i(v)$.
- [Case 2] If a node v is adjacent to $u \epsilon T_{bcast}$ then connect $u \rightarrow v$ and $v \rightarrow Cov_i(v)$.
- [Case 3] If a node $x \epsilon Cov_i(v)$ is in T_{bcast}, then connect $x \rightarrow v$ and $v \rightarrow Cov_i(v)$.

If there exists some nodes $v \epsilon V$ that cannot be connected to T_{bcast} using Step 1, then go to Step 2.

Algorithm 2 CSCA_MT Algorithm for Broadcast Tree Construction

Input: A duty cycled WSN $G=(V, E)$ and $s \epsilon V$

Output: Broadcast Tree T_{bcast}

```
procedure PROCESS STAGE1(G, s)
    S.push(s)
    while V is not empty do
      while S is not empty do
         v := S.pop()
         for i ϵ Ta(v) do
           if v ϵ Ci then
             v → Covi(v)
             S.push(Covi(v))
             Tbcast := Tbcast ∪ {v} ∪ Covi(v)
           end if
           if v is Adjacent to u ϵ Tbcast then
             u → v
             v → Covi(v)
             S.push(Covi(v))
             Tbcast := Tbcast ∪ {v} ∪ Covi(v)
           end if
           if xϵ Covi(v) is in Tbcast then
             x → v
             v → Covi(v)
             S.push(Covi(v))
             Tbcast := Tbcast ∪ {v} ∪ Covi(v)
           end if
         end for
         V = V \ {v}
      end while
      Process Stage2(V)
    end while
end procedure

procedure PROCESS STAGE2(V)
  if V is not empty then
    for v ϵ V do
      if x ϵ Covi(v) is adjacent u ϵ Tbcast then
        u → x
        x → v
        v → Covi(v)
        S.push(x)
        S.push(v)
        S.push(Covi(v))
        Tbcast := Tbcast ∪ {x} ∪{v} ∪ Covi(v)
        Return
      end if
    end for
  end if
end procedure
```

Algorithm 2. Pseudo code of CSCA_MT for Broadcast Tree Construction

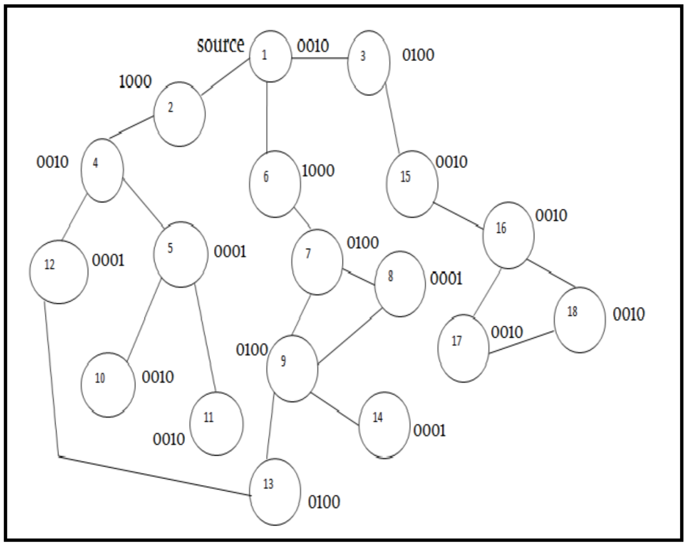

Figure 2. Duty-Cycled WSN G with 18 Nodes

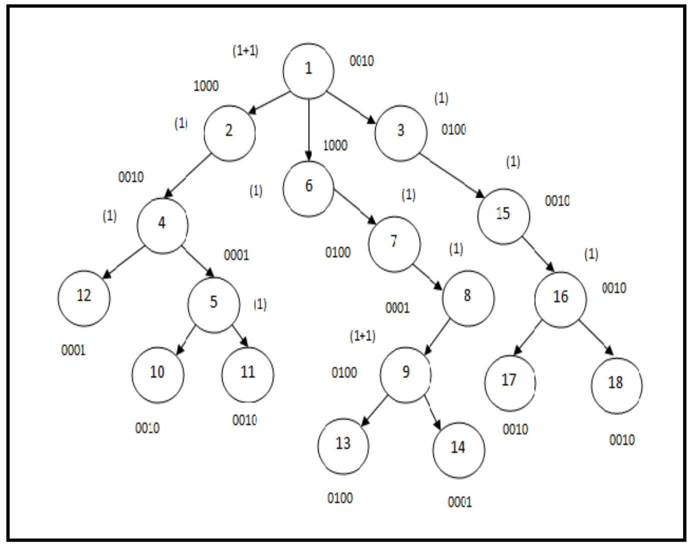

Figure 3. Broadcast Tree Constructed by CSCA with Cost(T) =13

Step 2:

- [Case 4] If a node $x \in Cov_i(v)$ is adjacent to some node u in T_{bcast}, then connect $u \rightarrow x$, $x \rightarrow v$ and $v \rightarrow Cov_i(v)$.

Go back to Step 1. Pseudo code for CSCA_MT is presented in Algorithm 2. The time complexity of Algorithm 2 is $O(N^3T)$ and complexity of Algorithm 1 is $O(N^2T)$ [1].

Fig. 1 illustrates the four conditions mentioned in Step 1 and 2. It is clear from Fig.1 that in case 1, a node v will require only 1 transmission, in case 2 and 3 v will require 2 transmissions, while in case 4 v will require 3 transmissions to be added to T_{bcast}. CSCA alogithm [1] arbitrarily selects any node v to be included into T_{bcast} that satisfies any of the four cases mentioned in Fig. 1. Our algorithm CSCA_MT gives priorities to the nodes satisfying case 1, 2 and 3 by deferring the nodes satisfying condition 4 to be connected to T_{bcast}. The aim is to reduce the number of nodes to be added into T_{bcast} with 3 transmissions. By adding the nodes satisfying case 1, 2 and 3 before including a node that require 3 transmissions, CSCA_MT reduces the number of data transmissions and it is evident from our experimental result.

Fig. 2 shows a duty-cycled WSN G consists of *18* nodes. Each node u is associated with a wakeup schedule of length *4* and is active in exactly one time slot. Fig. 3 and 4 shows the broadcast tree constructed by CSCA and CSCA_MT algorithm on G, respectively. The main difference is the way node *7* is connected into T_{bcast}. As shown in Fig. 3, CSCA algorithm connects *7* to $6 \in T_{bcast}$ and node *7* belongs to $Cov_2(8)$. Thus using case 4, CSCA connects $6 \rightarrow 7$, $7 \rightarrow 8$ and $8 \rightarrow Cov_2(8) = \{9\}$. The cost of T_{bcast} generated by CSCA is *13*. Whereas in Fig. 4, node *13* belongs to $Cov_2(9)$ and is adjacent to node $12 \in T_{bcast}$. Thus *13* is connected to *12* and connects *9* to it using case 4. Node *9* in turn connects *7* and *8* into T_{bcast} using only 1 transmission. The cost of T_{bcast} generated by CSCA_MT is 12. Thus by including all the nodes satisfying case 1, 2 and 3 before node *7*, CSCA_MT is able to reduce the cost of T_{bcast} by 1.

IV. Simulation Results

We conducted extensive simulations in order to the study the performance proposed CSCA_MT algorithm. We compared our results with the performance of existing CSCA [1] algorithm. The performance metrics used were the total number of data transmissions, the average height of broadcast tree and the average delay of broadcast operation.

We used Java platform (JDK 7) in all our simulation results. We generated WSNs by deploying the sensor nodes randomly in a geographical area of 200m×200m. We considered networks with 50, 100, 150, 200, 250, 300, 350 and 400 nodes. The number of node is denoted by N. For each value of N, we generated 1000 graphs. Our results are averaged over 1000 graphs for each value of N. A node u can transmit to a node v if and only if the Euclidean distance between u and v is less than or equal to the transmission range of sensor nodes.

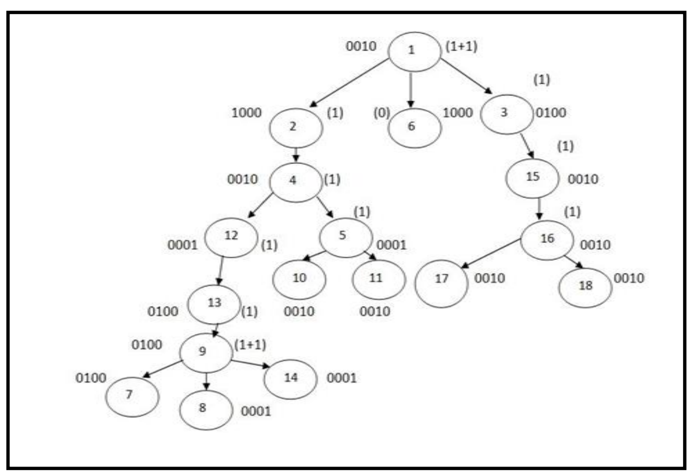

Figure 4. Broadcast Tree Constructed by CSCA_MT with Cost(T) =12

TABLE II. TOTAL NUMBER OF TRANSMISSIONS FOR NETWORKS WITH 50 TO 400 NODES

Number of Nodes N	**Total Number of Transmissions**	
	CSCA_MT	***CSCA***
50	25	27
100	49	53
150	72	78
200	92	101
250	112	129
300	128	156
350	138	177
400	145	201

TABLE III. AVERAGE HEIGHT OF THE BROADCAST TREE FOR NETWORKS WITH 50 TO 400 NODES

Number of Nodes N	**Average Height of Broadcast Tree**	
	CSCA_MT	***CSCA***
50	8	10
100	12	14
150	16	18
200	20	20
250	25	23
300	28	25
350	31	27
400	33	28

Given a node density of 8, transmission range of a sensor node is calculated as $\sqrt{(8 \times 200 \times 200)/(\pi \times N)}$. For each network, a source node is selected at random to initiate the broadcast message. Every node is associated with a wakeup schedule and in our experiments we used schedule length *sch_len* of 5. Every node in the network is active in only one randomly selected time slot. The results are presented in following sections.

A. Total Number of Data Transmissions

Our proposed CSCA_MT algorithm reduces the number of data transmissions as compared to CSCA algorithm. Total number of transmissions needed by CSCA_MT and CSCA algorithm is presented in Table II and is graphically plotted in Fig. 5. As shown in Fig. 5, for small size networks with 50 nodes we have almost same result as CSCA algorithm. As the number of nodes increases, CSCA_MT exhibits significant improvement in decreasing number of transmissions. For networks with 150 to 400 nodes, the differences in the result are more noticeable. CSCA_MT achieves an improvement of 27.86% and 22.03% at *N*=400 and *N*=350, respectively over CSCA in dropping number of data transmissions. This implies that for large networks there are considerable number of nodes satisfying condition 4 and thus are skipped by CSCA_MT to be connected later into the tree.

B. Average Height of Broadcast Tree

We calculated the height of broadcast trees generated by both CSCA_MT and CSCA algorithm and compared the result. For each value of *N*, the average height of broadcast tree is given in Table III and is graphically illustrated in Fig. 6. As shown in Fig. 6, for networks with 50 to 200 nodes CSCA_MT algorithm generates broadcast tree with less height as compared to that of CSCA algorithm. Then with the increase of number of nodes, CSCA_MT begins to generate broadcast trees with larger height. At *N*=350 and *N=400*, CSCA_MT increases the height of broadcast tree by 14.81% and 17.86% as compared to that of CSCA algorithm. For larger networks, more nodes satisfying condition 4 are skipped and connected later in tree

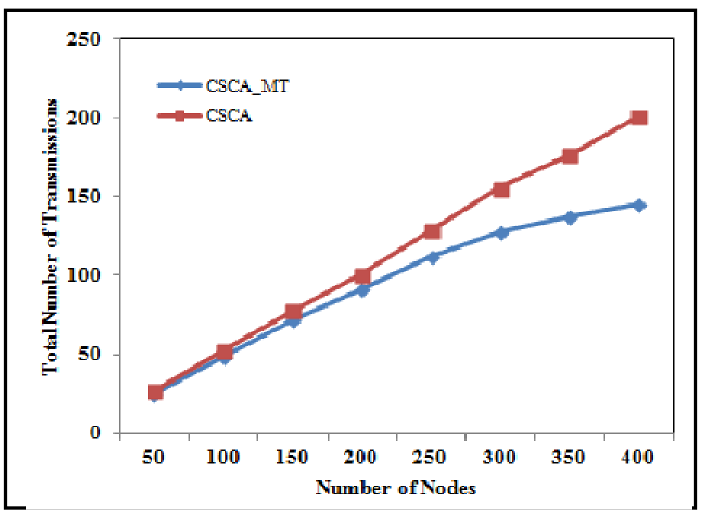

Figure 5. Total Number of Data Transmissions for CSCA and CSCA_MTAlgorithm

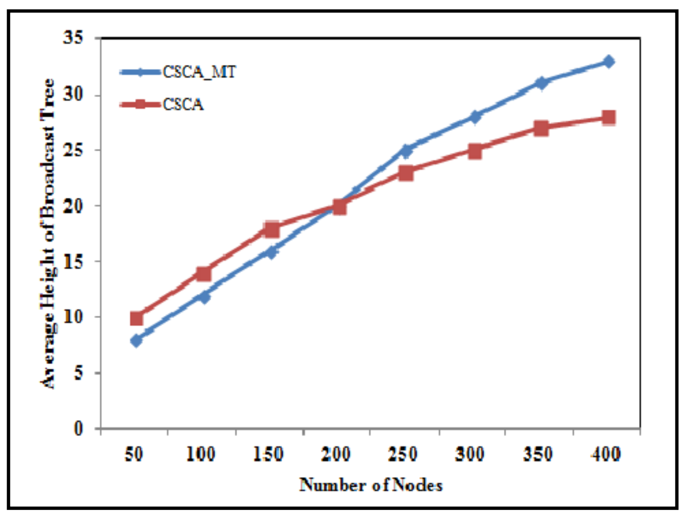

Figure 6. Average Height of Broadcast Tree Generated by CSCA and CSCA_MTAlgorithm

TABLE IV. AVERAGE DELAY OF BROADCAST OPERATION FOR NETWORKS WITH 50 TO 400 NODES

Number of Nodes N	Average Delay of Broadcast Operation	
	CSCA_MT	***CSCA***
50	15.68	15.36
100	22.83	21.67
150	27.35	26.57
200	32.76	31.19
250	36.98	35.11
300	40.86	37.96
350	43.35	40.50
400	45.65	42.95

and this will ultimately increase the height of broadcast tree. Broadcast tree with larger height will increase the number of hop counts that will in turn increase the network delay.

C. *Average Delay of Broadcast Operation*

Broadcast delay of a sensor node is defined as the total number of time slots a node will wait to transmit a data packet to all its neighbors after receiving that packet. From Table IV and Fig. 7 it is clear that CSCA_MT has almost same average delay as CSCA for networks with 50 to 250 nodes. For networks with more than 250 nodes, our algorithm produces more delay than the CSCA algorithm. For *N=400*, CSCA_MT produces about 6.28% more delay than that of CSCA algorithm. This is due to increase in the height of broadcast tree for networks with 250 to 400 nodes.

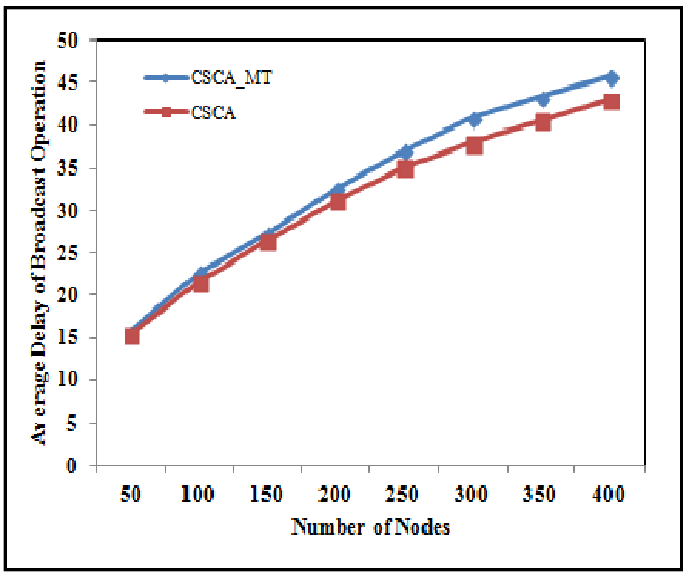

Figure 7. Average Delay of Broadcast Operation for CSCA and CSCA_MT

V. CONCLUSION

In this paper, we proposed a modification to the existing CSCA[1] algorithm and found that it improves the performance further by reducing the total number of data transmissions. Our proposed algorithm gives better result in terms of number of transmissions as compared to CSCA algorithm in both small and large size networks and good performance in terms of average height of broadcast tree and average delay of broadcast operation. We are currently working on developing more efficient broadcast algorithm that will reduce the tree height as well as average delay of the broadcast operation. Developing a distributed and localized algorithm to construct broadcast tree by utilizing the knowledge of 2-hop neighbors is also an interesting future work.

REFERENCES

[1] J. Hong, J. Cao, W. Li, S. Lu and D. Chen, "Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks", *IEEE Trans. Vehicular Technology*, vol. 59, no. 1, pp. 307-318, Jan. 2010.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A survey on sensor networks", *IEEE Communications Magazine*, Vol. 40, No. 8, pp.102-114, Aug. 2002.

[3] K. Han, Y. Liu and J. Luo, "Duty-cycle-aware minimum-energy multicasting in wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 21, no. 3, pp. 910 – 923, Jun. 2013.

[4] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, "Sleeping schedule aware minimum latency broadcast in wireless ad hoc networks," in *Proc. IEEE Intl. Conference on Communications (ICC'2009),* pp. 1–5, Dresden, Germany, Jun. 2009.

[5] Y. Duan, S. Ji, and Z. Cai, "Generalized broadcast scheduling in duty-cycle multihop wireless networks," in *Proc. 30th IEEE Intl. Conference on Performance Computing and Communications(IPCCC' 2011)*, pp. 1–6, Florida, USA, Nov. 2011.

[6] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *Proc. 28th IEEE Conf. Computer Communications (IEEE INFOCOM'2009)*, Rio De Janerio, Brazil, Apr. 2009, pp. 468–476.

[7] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in Proc. *21st Annual Joint Conference of theIEEE Computer and Communications Societies(INFOCOM'2002)*, vol. 3, pp. 1567–1576, 2002.

AUTHORS PROFILE

Rifatul Islam received BSc (Hons.) Degree in Computer Science and Engineering from University of Dhaka, Bangladesh in 2014. He is currently pursuing MSc. degree in Computer Science and Engineering in University of Dhaka. His field of interest includes wireless sensor network, image processing and software Architecture.

Umama Tasnim received B.Sc (Hons.) Degree in Computer Science and Engineering from University of Dhaka, Bangladesh in 2014. She is currently continuing her M.Sc degree in Computer Science and Engineering in University of Dhaka. Her research interest includes computer network, wireless sensor network and data mining.

Mosarrat Jahan received B.Sc (Hons.) and M.S degree from University of Dhaka, Bangladesh and M. Comp.Sc degree in Computer Science from Concordia University, Canada in 2012. Currently she is working as an assistant professor in Department of Computer Science and Engineering, University of Dhaka. Her research interest includes energy efficient data communications in wireless sensor networks, content management in mobile ad hoc networks and network security.

Shaily Kabir received B.Sc (Hons.) and M.S degree in Computer Science and Engineering from University of Dhaka, Bangladesh and also M.Comp.Sc degree in Computer Science from Concordia University, Canada in 2012. Currently she is working as an assistant professor in the Department of Computer Science and Engineering, University of Dhaka. Her research interests include computer networks and network security, data and web mining, and database management systems.