Computer Communications 149 (2020) 36–43

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Cluster Based Data Aggregation Scheme for Latency and Packet Loss Reduction in WSN

V. Seedha Devi [a,*], T. Ravi [b], S. Baghavathi Priya [c]

[a] Jaya Engineering College, Chennai, India
[b] Shadan Women's College of Engineering & Technology, Hyderabad, Telangana, India
[c] Rajalakshmi Engineering College, Thandalam, Chennai, India

ARTICLE INFO

Keywords:
Wireless Sensor Network (WSN)
Clustering
Data aggregation
Compressive sensing
Slot scheduling

ABSTRACT

In Wireless Sensor Networks (WSN), the main issues of cluster based data aggregation algorithms are energy balancing, packet loss and latency reduction. In existing scheduling algorithms for data aggregation, time slots are mostly assigned based on the data sensing period and data transmission rate, ignoring packet loss and latency. In this paper, Cluster based Data Aggregation Scheme for Latency and Packet Loss Reduction in WSN is proposed. The proposed scheme consists of two phases: Aggregation Tree Construction and Slot scheduling algorithm. In phase-1, each cluster head applies compressive aggregation for the data received from its members. Then the aggregation tree is constructed by the sink using Minimum Spanning Tree (MST). In phase-2, the packet loss rate and latency are taken into consideration while prioritizing and assigning timeslots to the nodes with aggregated data. This scheme avoids using unnecessary retransmissions and waiting, which results to be beneficial in enhancing the network performance in WSN. Simulation results show that the proposed scheme reduces the latency and overhead and increases the packet delivery ratio and residual energy.

1. Introduction

WSNs are basically deployed over large areas in a distributed manner to estimate the variation in physical as well as environmental factors in terms of pressure, temperature, sound, etc. In WSN, the sensor nodes interact with one another in order to transfer the data packets to the destination. Now WSN have evolved and performs its operation in a bidirectional manner which also aids in handling the sensor functioning in ease. The sensor nodes in WSN possess computational as well as communication ability and is hence more reliable when considered in comparison with the traditional wired network in unpredictable environments [1].

WSN is made up of minute sensor nodes which are smart enough to detect any change in the environmental conditions through its sensor units. Then this detected variation is sent to the sink through the wireless network. The WSN gather the required network data, operate on it and then transmit the determined information to the corresponding destination. The sensor nodes in WSN are distributed over large geographical areas in huge numbers in all places including the ones which cannot be handled by humans. So, it is necessary to ensure that the energy usage of the nodes is limited and are deployed in a non uniform manner. Also, it is important for the sensor nodes to co-operate with one another in WSN in order to maintain the self organizing nature of the network. The modules used in sensor nodes also need to be effective so as to consume minimum energy to assure maximized lifetime for the network [2]. WSN is employed in many applications like military surveillance, environment monitoring, exploration, etc. In almost all of the applications, the WSN aggregates the data that has been sensed by its nodes and then deliver it to its sink, where it gets examined and investigated. The sensed data is usually delivered at the sink through multi-hop routing. The sink stores the received data at the database, then operates on it using the control commands and investigates the data to determine the network event [3].

Data aggregation is a process of minimizing the transmission count of the data packets being transmitted using the in-network data processing [4]. It is a data transmission mechanism which intakes data transmitted from various nodes and forms a single packet after identifying and removing the redundant packets. This process reduces the transmission count and in turn consumes lesser energy. Several data aggregation mechanisms have been proposed and the selection of the mechanism should be made on the basis of the application needs and energy usage involved. The main data aggregation mechanisms are Cluster based data aggregation and Tree based data aggregation [5].

In cluster based data aggregation, the Cluster Head (CH) aggregates the data sent by its members and then transmits the aggregated data to

* Corresponding author.
E-mail addresses: seethaitjec@gmail.com (V. Seedha Devi), travi675@yahoo.com (T. Ravi), baghavathipriya.s@rajalakshmi.edu.in (S.B. Priya).

https://doi.org/10.1016/j.comcom.2019.10.003
Received 4 July 2019; Received in revised form 19 August 2019; Accepted 2 October 2019
Available online 4 October 2019
0140-3664/

the Base Station (BS) [6]. The CH aggregates the data after decreasing the size of the packet and redundancy. Later the packet gets transferred to the destination [1]. In hierarchical network, the effectiveness of data aggregation technique can be maximized by employing the clustering concept. Clustering can be performed in two ways: static clustering and dynamic clustering. In static clustering, the network is grouped into many smaller clusters. Energy consumption in an unpredictable manner is a very common and serious issue in cluster based WSN. In tree based data aggregation, the sink node act as root node along with leaves assumed as source node. In this technique data is transfer from leaves node towards sink and aggregation is performed through parent node. In WSN, few nodes may exhaust its energy sooner and lead to network failure. To overcome this issue, several researchers have analyzed and presented new algorithms [6].

The main issue of cluster based aggregation algorithms are energy balancing and latency reduction. In compressive sensing (CS) based aggregation schemes [7–9], if recovery of original packets fails due to incorrect compression ratio, it may lead to packet losses. Moreover, packet losses may occur due to congestion or bad channel conditions which may not be resolved by using CS.

In existing scheduling algorithms [10,11], time slots are calculated based on the data sensing period and data transmission rate. For real-time data with high priority, data loss and delay are crucial parameters which have impact on scheduling. Hence time slot assignment should consider these two parameters in order to ensure reliable delivery of critical and real-time data.

Hence, the main objective of this work is to design a scheduling algorithm for cluster based aggregation which should provide

- balanced energy consumption
- reduced end-to-end latency
- reduced packet loss rate

To meet this objective, we propose to develop a cluster based aggregation scheduling scheme for latency and packet loss reduction in WSN.

The paper is organized as follows. Section 2 presents the literature review of existing works. Section 3 presents the proposed cluster based data aggregation scheduling scheme in detail. Section 4 presents the experimental results and discussion. The paper is concluded in Section 5.

2. Related works

Dnyaneshwar S. Mantri et al. [4] have presented a Bandwidth Efficient Cluster-based Data Aggregation (BECDA) algorithm. This algorithm offered a solution for efficient data gathering for in-network aggregation. This algorithm takes into consideration a network consisting of heterogeneous sensor nodes w.r.t energy and dynamic sink nature inorder to fuse the data packets. The optimized technique to attain the target is to employ intra and inter cluster aggregation on the sensor nodes that are distributes across the network in a random manner at varying rate. This technique utilizes data correlation within the data packet to deploy the aggregation function on the data. A set of data aggregation scheduling algorithms has been proposed for energy balancing and latency reduction.

Miloud Bagaa et al. [10] have presented a Distributed algorithm for Integrated tree Construction and data Aggregation (DICA). This algorithm comprises of tree formation along with node scheduling mechanism in order to minimize the latency involved. In this algorithm, selection of parent node can be performed in a variety of ways. Also, this technique ensure minimized latency as well as maximized reuse of time slots.

Sain Saginbekov et al. [11] have presented a distributed data aggregation scheduling (DAS) algorithm for Wireless Sensor Networks which possess two sinks. In this paper, a distributed energy balancing algorithm is also presented which aims at balancing the energy that is being consumed by the cluster head for aggregation purpose. In this algorithm, initially a tree which is rooted at nodes is created and is referred as virtual sinks. Next the children count at each level is managed. Later, the developed tree is taken into consideration by the DAS algorithm which allots timeslots to the consecutive nodes in order to minimize the redundant wastage of energy that may be caused due to repeated active-sleep transitions. Apart from the traditional static equal clustering algorithms, a set of unequal and dynamic clustering algorithms has been proposed.

Jun YUE et al. [12] have presented a now unequal cluster-based data aggregation protocol. In this proposed protocol, the network is divided into different sized grids. Then the cluster head rotation is followed in every grid in the network. This technique is capable of energy consumption balancing through applying appropriate grid size in order to manage the number of nodes which get involved in cluster head rotation from various grids. Also, in this technique, the energy efficiency is improved due to the some external schemes.

Woo-Sung Jung et al. [13] have presented a hybrid clustering technique on the basis of data aggregation mechanism. In this technique, any appropriate clustering process can be selected and employed on the basis of the network status, data aggregation effectiveness enhancement along with energy usage and data transmission ratio.

In order to reduce the aggregation overhead and energy consumption, compressive data aggregation techniques have been proposed. For energy-constrained WSNs, Compressive Sensing (CS) provides an effective data gathering approach.

Cuicui Lv et al. [7] have developed a mobile agent based compressive data gathering algorithm (MA-Greedy algorithm). In this algorithm, measurement matrix which is referred as sparse binary matrix is employed in CS. Coefficient of Variation(CV) metric is developed to assess the balance level achieved with the energy utilization in the nodes.

Dariush Ebrahimi et al. [8] have presented a decentralized technique for the compressive data gathering problem (DCDG). In this technique, every node is responsible for making a decision regarding the construction and maintenance of the forwarding trees. Spare projections are designed to pre-process the sensed data in the WSN. A projection is basically a compressed packet which consists of different packets from various nodes. After every projection is gathered, the sink solves the convex optimization problem in order to retrieve the original data.

Xuangou Wu et al. [9] have proposed a sparsest random scheduling technique for compressive data gathering mechanism. This technique consists of a sparsest measurement matrix in which every row consists of just a single non zero entry. A representation basis scheme is developed on the basis of the sensory data along with sparsest measurement matrix so as to retrieve the sensory data precisely and also to minimize the cost involved in data transmission.

3. Cluster based Data Aggregation Scheme for latency and packet loss reduction

3.1. Overview

In this paper, we propose to develop a cluster based aggregation scheduling scheme for latency and packet loss reduction in WSN. Initially, the network is clustered and cluster heads are elected using Artificial Bee Colony (ABC) algorithm as described in our previous work [14].

The proposed scheme consists of two phases:

Phase-1: Aggregation Tree Construction
Phase-2: Slot scheduling algorithm

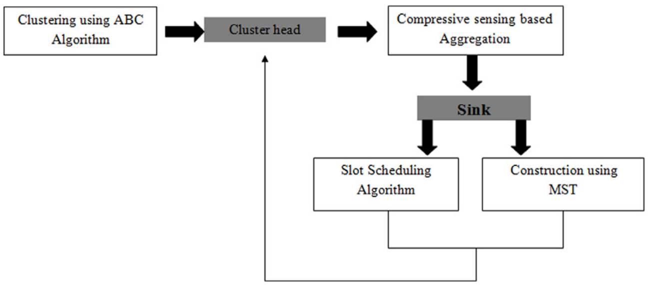

Fig. 1. Architecture of proposed scheme.

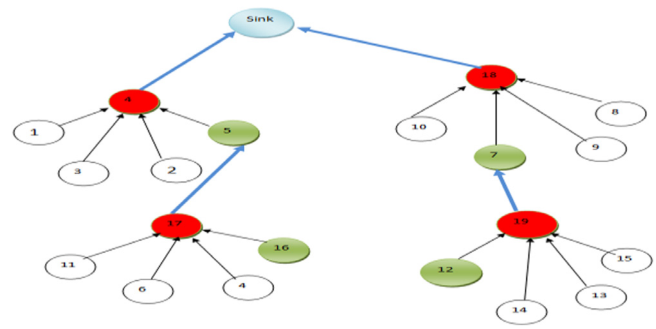

Fig. 2. Cluster based aggregation tree.

In phase-1, each CH aggregates the received data from its members using perfectly compressive aggregation function [4] in which the correlation between the generated packets is considered. After cluster formation, the CH broadcasts its ADV, including the CH ID, location, cluster ID, cluster size and residual energy. The sink based on the position of CHs, provides a minimum spanning tree (MST) between them and finally broadcast the tree information for all CHs [5]. It also includes a schedule for data transmission among the cluster heads.

In phase-2, the expected packet loss rate (PLR) is estimated at the sink. For delay estimation, the effective maximum delay (dm) at each node is estimated [15]. For nodes with real-time data having delay and loss constraints, time slots are allocated based on PLR and dm. For other nodes, time slots are assigned based on transmission rate.

Since the packet scheduling avoids losses and meets deadline, the energy consumption due to unnecessary waiting and retransmission can be avoided. Fig. 1 shows the architecture of the proposed scheme.

3.2. Aggregation Tree Construction

Initially, clusters are formed and CHs are selected as described in our previous paper [14]. Each cluster has N cluster members (CM) which generates variable number of data packets of fixed size. In Aggregation Tree Construction phase, the cluster head aggregates all the data packets received by it from its cluster members through the compressive aggregation function. The Sink forms the Minimum Spanning Tree (MST) using the information collected by it from all the cluster head in the network. This MST is later used by the sink to determine the distance between the different cluster heads and then to schedule the data transmission for every cluster. This process is described in algorithm 1 (see [16]).

CH_j broadcasts its cluster details consisting of its CH_{ID}, CH_{loc}, C_{ID}, C_{size}, and E_{res}, to its CM as well as to its neighboring CH_{jk}. Then CM_i transmits its data packets to its CH_j. CH_j calculates the data reception rate (rate at which data packets are received) from its cluster members using Eq. (1). CH_j applies compressive aggregation function in order to aggregate the received data packets using Eq. (2). Each CH transmits its data and CH_{loc} to BS. On receiving the CH_{loc} from every CH, BS records this information in its database. Based on each CH_{loc}, BS creates a MST between all the CH. It selects the minimum distance from the vertex of one tree to another tree vertex on the basis of the CH_{loc}. This details regarding to the minimum distance between two clusters is used by the BS to schedule data transmission. The MST along with the data transmission schedule for each cluster is broadcast by the BS to all CHs.

Thus, the sink links all the clusters in the network through the Minimum Spanning Tree which aids in seamless data transmission later. The cluster head will store the received data transmission schedule details from the Base Station and operate during its schedule.

Fig. 2 describes the cluster-based aggregation tree architecture. It is constructed by taking sink as the root and sensor nodes as the leaves connected by CHs and gateways.

Table 1
Simulation parameters.

Number of nodes	50 to 100
Network size	1000 × 1000 m
MAC protocol	IEEE 802.11
Transmission rate	50 to 250 kb/s
Data flows	2 to 10
Antenna model	Omni antenna
Initial energy	14.0 J
Transmission power	0.660 W
Receiving power	0.4 W
Packet size	512 bytes

3.3. Slot scheduling algorithm

In Phase 2, the CH prioritizes the aggregated data so that it can be delivered at the sink without failing to fulfill the delivery constraints. The BS allots timeslot for the data based on the priority of data. This process is described in algorithm 1.

If the data is real time data with strict deadlines and by BS and the timeslots are allotted, such that the data packet gets delivered at the sink within reliability constraints, then the data is given high priority. If the data is non real-time with lesser constraints, then the data is given low priority. For the high priority data, the PLR value and D are considered the D and PLR level. BS allots timeslot for the low priority data transmission based on the sensing period and R_{txn}. When the data transmission time scheduled for a CH by BS arrives, the CH initiates transmission of the higher priority data followed by the lower priority data in the allotted timeslots.

In this way, the packet loss issue can reduced to a greater extent and the packet delivery at the sink can be met within the deadline. In this process, since there is no redundant waiting prior data transmission, the energy consumption is also controlled.

4. Simulation results

The proposed Cluster based Data Aggregation Scheme (CDAS) is simulated using NS-2. The proposed CDAS scheme is compared with Bandwidth Efficient Cluster-based Data Aggregation (BECDA) scheme [4]. The performance metrics packet delivery ratio packet drop, average residual energy and control overhead are measured for these two schemes. The Simulation settings in Table 1 is shown.

4.1. Results and discussion

4.1.1. Varying the data flows

In order to analyze the effect of increased data flows, the number of data flows across each cluster is varied from 2 to 10. The data flows consist of equal number of real-time and non real-time traffic. The number of nodes is fixed as 200.

The result of end-to-end delay for the two schemes is shown in Fig. 3. As the data flows are increased, the waiting time at aggregation

Algorithm 1

Notations	Meaning
CH_j	j^{th} Cluster Head
CH_{jk}	neighbour cluster heads of CH_j
CM	Cluster Member
CH_{ID}	Identity of the Cluster Head
CH_{loc}	Location of the Cluster Head
C_{ID}	Cluster ID
C_{size}	Cluster Size
E_{res}	Residual Energy
DR_{RXT}	Data packet reception rate
ER	symbol error rate
n_i	node
MST	Minimum Spanning Tree
n	number of nodes in a cluster
k	bits of data packets
M	data packet bits
$f(A_{CH})$	compressive aggregation function
X_i, Y_j	variables representing correlation between the number of packets generated by cluster members
V^1 and V^2	vertex of tree T1 and T2, respectively.
$Dist_{min}(V^1, V^2)$	minimum distance between V^1 and V^2
BS	Base Station

1. For each CH_j , j=1,2....n
   $\{CH_{id}, CH_{loc}, C_{ID}, C_{size}, E_{res}\}$
2. $CH_j \longrightarrow CM_i, CH_{jk}$
3. $CM_i \xrightarrow{\{Packets\}} CH_j$
4. CH_j estimates the packet generation rate DR_{RXT} using equation (1) [18]:
5. $$DR_{RXT}(i) = \prod_{j=1}^{2l}(1 - ER(i,j)) \tag{1}$$
6. CH_j applies compressive aggregation function f(A) at CH_j using equation (2) [4]:
7. $$f(A_{CH}) = \sum_{i=1}^{k} X_i + \frac{1}{M}\sum_{j=1}^{M} Y_j \tag{2}$$
8. CH_j transmits its data and CH_{loc} to BS
   $CH_j \longrightarrow CH_{loc} \longrightarrow BS$
9. If(BS receives CH_{loc} from CH_j) then
10. BS stores CH_{loc} in its table.
11. BS creates MST between all CH based on CH_{loc}
12. End if
13. BS selects $Dist_{min}(V^1, V^2)$ based on CH_{loc}
14. BS schedules the data transmission for CH_j using $Dist_{min}(V^1, V^2)$
15. BS broadcasts MST and schedule to CH_j
16. End For

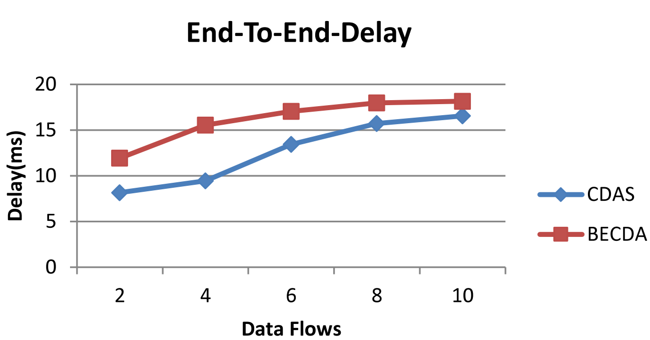

Fig. 3. Data flow vs. delay.

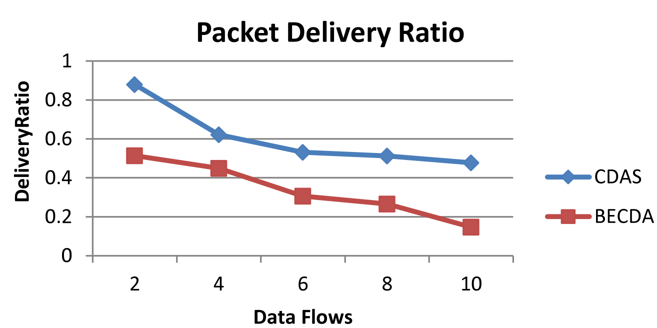

Fig. 4. Data flows vs. packet delivery ratio.

point increases, leading to increase in delay. The delay of CDAS increases from 8.1 ms to 16.5 ms and the delay of BECDA increases from 11.9 ms to 18.1 ms. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected delay of real-time flows. Moreover, the MST used in inter-cluster transmission reduces the transmission delay. Hence the delay of CDAS is 22% lesser when compared to BECDA.

The result of packet delivery ratio for the two schemes is shown in Fig. 4. As the data flows are increased, the queue at CH is overloaded, leading to decrease in delivery ratio. The delivery ratio of CDAS decreases from 0.87 to 0.47 and the delivery ratio of BECDA decreases from 0.51 to 0.14. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the delivery ratio of CDAS is 45% higher than BECDA.

Algorithm 2

Notations	Meaning
CH_j	Cluster Head
PLR	Packet Loss Rate
n	sensor node
j	Neighbor node of node i
D	packet's end-to-end delay
d_j	actual delay that packet experiences at node j
R_{txn}	Data Transmission Rate
P_{pri}	Priority of data packet

For all CH_j, j=1,2...n
    CH_j maintains all the aggregated data.
    CH_j examines the nature of each data within the aggregated data.
    If (data is real-time) then
        P_{pri} = High
    Else
        P_{pri} = Low
    End if
    CH_j estimates the D related to the data transmission using equation (3) [16]:

$$D = \sum_{j=1}^{i-1} \tilde{d}_j + \sum_{j=i+1}^{n} \tilde{d}_j \tag{3}$$

    CH_j estimates R_{txn} of the data packet.
    CH_j transmit aggregated data along with D, R_{txn} and P_{pri}
End For
For each aggregated data from CH_j,
    BS checks P_{pri} .
    If (P_{pri} = High) then
        BS estimates the PLR
        BS allocates time slots to CH_j such that delay < D and PLR = PLR_{min}
    Else
        BS allocates time slots to CH_j based on sensing period and R_{txn}
    End if
    BS transmits the allocated time slots to CH_j
End For
CH initiates data transmission in the allotted time slots.

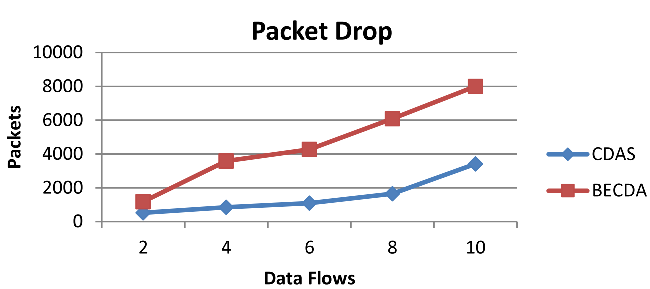

Fig. 5. Data flows vs. packet drop.

The result of average packet drop for the two schemes is shown in Fig. 5. As the data flows are increased, the queue at CH is overloaded, leading to increase in packet drop. The packet drop of CDAS increases from 522 to 3415 and the packet drop of BECDA increases from 1180 to 7999. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the drop of CDAS is 67% less when compared to BECDA.

The result of normalized overhead for the two schemes is shown in Fig. 6. As the data flows are increased, the aggregation overhead increases, leading to increase in normalized overhead. The overhead of CDAS increases from 0.09 to 0.4 and the overhead of BECDA increases from 0.23 to 0.68. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the overhead at CH. Hence the overhead of CDAS is 46% less when compared to BECDA.

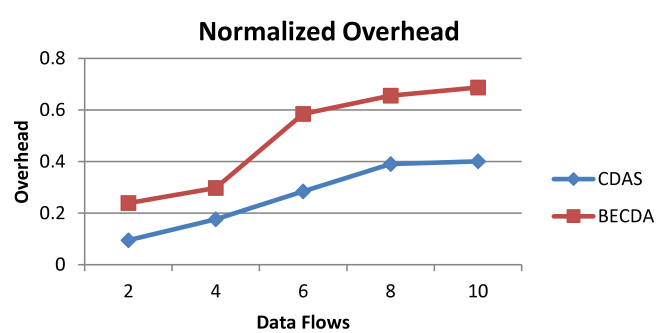

Fig. 6. Data flows vs. overhead.

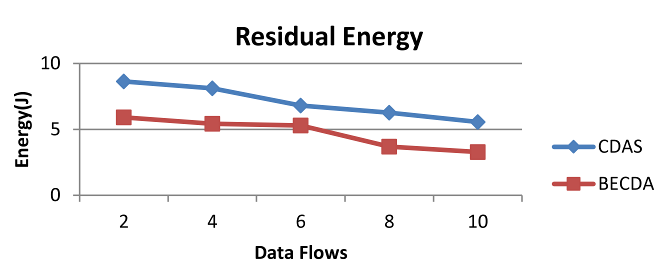

Fig. 7. Data flows vs. residual energy.

The result of average residual energy for the two schemes is shown in Fig. 7. As the data flows are increased, the residual energy of corresponding CH reduces leading to decrease in average residual energy.

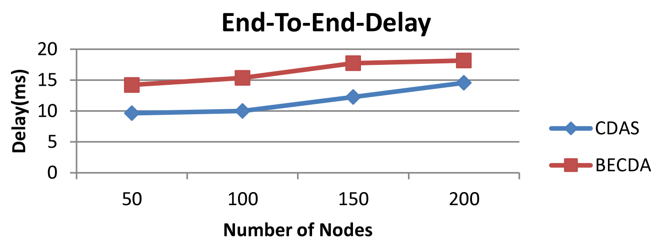

Fig. 8. Number of nodes vs. end-to-end delay.

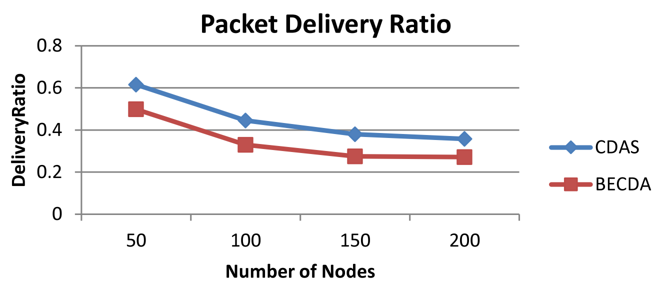

Fig. 9. Number of nodes vs. packet delivery ratio.

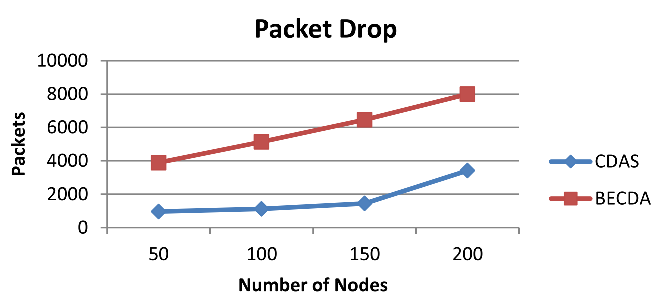

Fig. 10. Number of nodes vs. packet drop.

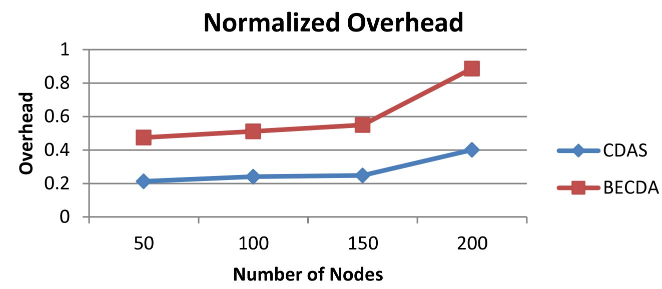

Fig. 11. Number of nodes vs. normalized overhead.

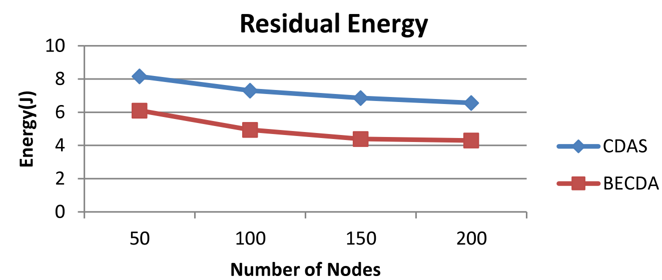

Fig. 12. Number of nodes vs. residual energy.

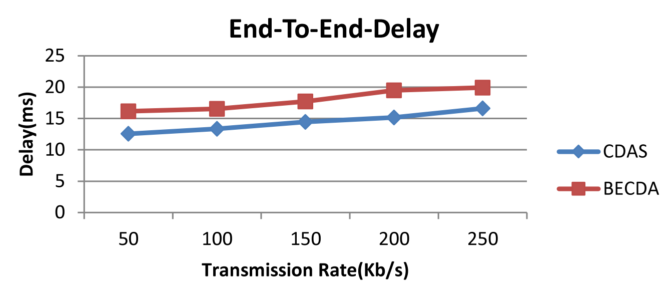

Fig. 13. Transmission rate vs. delay.

The residual energy of CDAS decreases from 8.63 to 5.56 J and the residual energy of BECDA decreases from 5.9 to 3.2 J. Since CDAS uses MST for inter-cluster routing and slot scheduling for transmission, the residual energy of CDAS is 33% higher than BECDA.

4.1.2. *Varying the nodes*

In order to analyze the effect of network size and node density, the number of nodes is varied from 50 to 200. The number of data flows is fixed as 4.

The result of end-to-end delay for the two schemes is shown in Fig. 8. As the number of nodes is increased, the length of the routing path increases, leading to increase in delay. The delay of CDAS increases from 9.6 ms to 14.5 ms and the delay of BECDA increases from 14.2 ms to 18.1 ms. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected delay of real-time flows. Moreover, the MST used in inter-cluster transmission reduces the transmission delay. Hence the delay of CDAS is 29% lesser when compared to BECDA.

The result of packet delivery ratio for the two schemes is shown in Fig. 9. As the number of nodes is increased, packet drop at intermediate hops increases, leading to decrease in delivery ratio. The delivery ratio of CDAS decreases from 0.61 to 0.35 and the delivery ratio of BECDA decreases from 0.49 to 0.27. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the delivery ratio of CDAS is 24% higher than BECDA.

The result of average packet drop for the two schemes is shown in Fig. 10. As the number of nodes is increased, packet drop at intermediate hops increases. Hence the packet drop of CDAS increases from 963 to 3415 and the packet drop of BECDA increases from 3894 to 7999. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the packet drop of CDAS is 72% less when compared to BECDA.

The result of normalized overhead for the two schemes is shown in Fig. 11. As the number of nodes is increased, the aggregation overhead is increases, leading to increase in normalized overhead. The overhead of CDAS increases from 0.21 to 0.40 and the overhead of BECDA increases from 0.47 to 0.88. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the overhead at CH. Hence the overhead of CDAS is 54% less when compared to BECDA.

The result of average residual energy for the two schemes is shown in Fig. 12. As the data number of nodes is increased, the residual energy corresponding nodes leading to decrease in average residual energy. The residual energy of CDAS decreases from 8.1 to 6.5 J and the residual energy of BECDA decreases from 6.1 to 4.2 J. Since CDAS uses MST for inter-cluster routing and slot scheduling for transmission, the residual energy of CDAS is 32% higher than BECDA.

4.1.3. *Varying the transmission rate*

In order to analyze the effect of increased data transmission rate, the transmission rate is varied from 50 to 250 kb/s. The data flows consist of equal number of real-time and non real-time traffic. The number of nodes is fixed as 200.

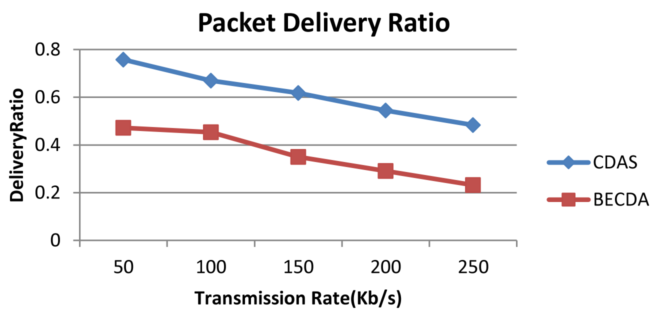

Fig. 14. Transmission rate vs. packet delivery ratio.

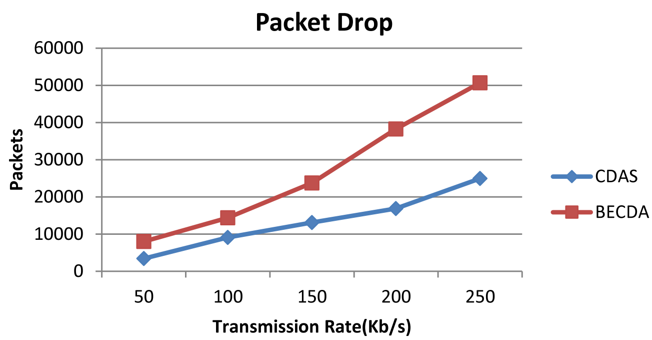

Fig. 15. Transmission rate vs. packet drop.

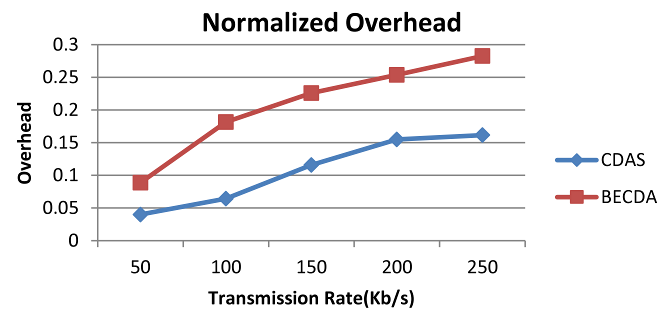

Fig. 16. Transmission rate vs. overhead.

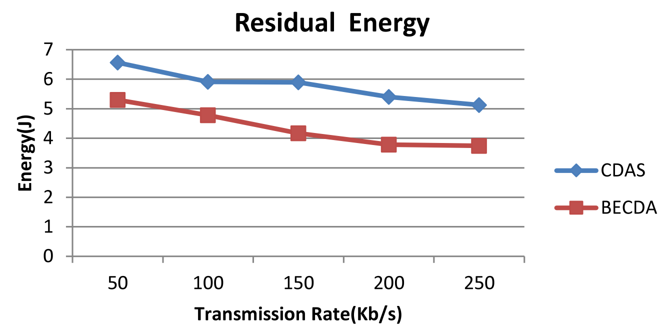

Fig. 17. Transmission rate vs. residual energy.

The result of end-to-end delay for the two schemes is shown in Fig. 13. As the transmission rate is increased, the waiting time at aggregation point increases, leading to increase in delay. The delay of CDAS increases from 12.5 ms to 16.6 ms and the delay of BECDA increases from 16.1 ms to 19.9 ms. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected delay of real-time flows. Moreover, the MST used in inter-cluster transmission reduces the transmission delay. Hence the delay of CDAS is 19% lesser when compared to BECDA.

The result of packet delivery ratio for the two schemes is shown in Fig. 14. As the transmission rate is increased, the queue at CH is overloaded, leading to decrease in delivery ratio. The delivery ratio of CDAS decreases from 0.75 to 0.48 and the delivery ratio of BECDA decreases from 0.47 to 0.23. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the delivery ratio of CDAS is 45% higher than BECDA.

The result of average packet drop for the two schemes is shown in Fig. 15. As the rate is increased, the queue at CH is overloaded, leading to increase in packet drop. The packet drop of CDAS increases from 3415 to 24 957 and the packet drop of BECDA increases from 7999 to 50 675. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the expected packet loss rate of real-time flows. Hence the drop of CDAS is 49% less when compared to BECDA.

The result of normalized overhead for the two schemes is shown in Fig. 16. As the data transmission rate is increased, the aggregation overhead increases, leading to increase in normalized overhead. The overhead of CDAS increases from 0.04 to 0.16 and the overhead of BECDA increases from 0.08 to 0.28. Though both the schemes apply the aggregation function at CH, CDAS uses the slot scheduling algorithm, which reduces the overhead at CH. Hence the overhead of CDAS is 50% less when compared to BECDA.

The result of average residual energy for the two schemes is shown in Fig. 17. As the data transmission rate is increased, the residual energy of corresponding CH reduces leading to decrease in average residual energy. The residual energy of CDAS decreases from 6.5 to 5.1 J and the residual energy of BECDA decreases from 5.2 to 3.7 J. Since CDAS uses MST for inter-cluster routing and slot scheduling for transmission, the residual energy of CDAS is 24% higher than BECDA.

5. Conclusion

In this paper, we have proposed a Cluster based Data Aggregation Scheme for Latency and Packet Loss Reduction in WSN. The processing of this scheme is divided into two phases: Aggregation Tree Construction phase and Slot scheduling algorithm. In the first phase i.e., Aggregation Tree Construction phase, the data packets received from the cluster members is aggregated by the cluster head using compressive aggregation function. The Base Station creates a Minimum Spanning Tree (MST) based on every cluster information and schedules the data transmission time for each cluster head. In the second phase i.e., the slot scheduling algorithm, the cluster head classifies its aggregated data into high and low priority data. The high priority data is assigned time slot on a prioritized basis, whereas the low priority data are enqueued and are allotted timeslots after serving the high priority data. In this way, packet loss is minimized and this ensures timely packet delivery. The Proposed system reduces the overhead and end-to-end delay. This proposed process also optimizes energy consumption by the nodes since packet retransmission and unnecessary waiting is avoided in this process and the lifetime of the network is increased.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Shivani Desai, Anuja R. NairTarjni Vyas, Hunar Jain, Survey on cluster based data aggregation in wireless sensor network, Int. J. Adv. Res. Comput. Sci. 8 (5) (2017).

[2] Soundaram Jothi, Muthial Chandrasekaran, Cluster based compressed data aggregation and routing in WSN, Int. J. Intell. Eng. Syst. 9 (4) (2016).

[3] Cheng Feng, Zhijun Li, Shouxu Jiang, Weipeng Jing, Delay-constrained data aggregation scheduling in wireless sensor networks, Int. J. Distrib. Sens. Netw. 13 (6) (2017).

[4] Dnyaneshwar S. Mantri, Neeli Rashmi Prasad, Ramjee Prasad, Bandwidth efficient cluster-based data aggregation for Wireless Sensor Network, Comput. Electr. Eng. (2014).

[5] Hedieh Sajedi, Zahra Saadati, A hybrid structure for data aggregation in wireless sensor network, J. Comput. Eng. 2014 (2014) Article ID 395868.

[6] Jun Yuea, Weiming Zhang, Weidong Xiao, Daquan Tang, Jiuyang Tang, Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks, in: International Workshop on Information and Electronics Engineering (IWIEE), 2012.

[7] Cuicui Lv, QiangWang, WenjieYan, YiShen, Energy-balanced compressive data gathering in Wireless Sensor Networks, J. Netw. Comput. Appl. 61 (2016) 102–114.

[8] Dariush Ebrahimi, Chadi Assi, A distributed method for compressive data gathering in wireless sensor networks, IEEE Commun. Lett. 18 (4) (2014).

[9] Xuangou Wu, Yan Xiong, Panlong Yang, Shouhong Wan, Wenchao Huang, Sparsest random scheduling for compressive data gathering in wireless sensor networks, IEEE Trans. Wireless Commun. 13 (10) (2014).

[10] Miloud Bagaa, Mohamed Younis, Djamel Djenouri, Abdelouahid Derhab, Nadjib Badache, Distributed low- latency data aggregation scheduling in wireless sensor networks, ACM Trans. Sensor Netw., Vol. 11, No. 3, Article 49.

[11] Sain Saginbekov, Arshad Jhumka, Many-to-many data aggregation scheduling in wireless sensor networks with two sinks, Comput. Netw. 123 (2017) 184–199.

[12] Jun Yue, Weiming Zhang, Weidong Xiao, Daquan Tang, Jiuyang Tang, A Novel Unequal Cluster-Based Data Aggregation Protocol for Wireless Sensor Networks, Przegląd Elektrotechniczny, ISSN 0033-2097.

[13] Woo-Sung Jung, Keun-Woo Lim, Young-Bae Ko, Sang-Joon Park, A Hybrid Approach for Clustering-based Data Aggregation in Wireless Sensor Networks, IEEE, 2009.

[14] V. Seedha Devi, Dr T. Ravi, Cluster head election algorithm in wireless sensor network using artificial bee colony, Asian J. Res. Soc. Sci. Hum. 6 (10) (2016).

[15] Lin Tang, Quansheng Guan, Shengming Jiang, Bingyi Guo, A deadline-aware and distance-aware packet scheduling algorithm for wireless multimedia sensor networks, Hindawi Publishing Corporation International Journal of Distributed Sensor Networks, Vol. 2015.

[16] Zhiwei Zhao, Xianghua Xu, Wei Dong, Jiajun Bu, An accurate link correlation estimator for improving wireless protocol performance, Sensors (ISSN: 1424-8220) (2015) www.mdpi.com/journal/sensors.