Paper

ICBCA – Improved Cluster Based Channel Allocation in Cognitive Radio Sensor Networks

Jayashree Agarkhed and Veeranna Gatate

Computer Science and Engineering Department, Poojya Doddappa Appa College of Engineering, Kalaburagi, Karnataka, India

https://doi.org/10.26636/jtit.2020.139820

Abstract—Wireless sensor networks (WSNs) operate in an overcrowded electromagnetic environment, as the spectrum is shared by various wireless communication technologies. This gives rise to various challenges related to optimized and efficient spectrum utilization. Cognitive radio (CR) has emerged as a solution satisfying this requirement, as it is capable of adapting to the dynamic radio spectrum. Thanks to the deployment of cognitive radio in WSNs, the spectrum may be utilized in a more efficient manner. CR may identify the vacant channels dynamically, allowing the sensor nodes to effectively communicate with each other. In this paper a clustering algorithm known as improved cluster-based channel assignment (ICBCA) is implemented, forming clusters of CR sensor nodes and then selecting vacant channels for data transmission purposes. Simulation results show that ICBCA outperforms existing clustering algorithms in CR sensor networks.

Keywords—channel interference, channel overlapping, co-channel interference, cognitive radio.

1. Introduction

Wireless sensor networks (WSNs) are a collection of battery powered nodes used to monitor the environment. These networks are successfully applied in monitoring health, environment and habitat monitoring, in military, surveillance and weather applications, as well as in disaster management [1], [2]. All sensor nodes forward the data to the base station (BS). WSNs are often deployed in usually unattended areas where physical presence of human beings is difficult to ensure due to practical reasons. Therefore, the replacement of batteries used by such networks becomes a major energy preservation-related concern [1]. Grouping sensor nodes into clusters may be a remedy enabling to preserve energy in WSNs [3]. Here, the sensor nodes of a network are controlled by a centralized coordinator, known as the cluster head (CH). All nodes within the cluster communicate with the CH forwarding the sensed data. The CH, in turn, communicates with its neighboring CH and this process is continued until the sensed data reach the BS. Research focusing on energy conservation in WSNs is presented in [4] and [5].

Due to advances in technology, more wireless solutions are becoming available everyday, resulting in the excessive used of the spectrum. The radio spectrum is divided into a licensed band, allocated to licensed users referred to as primary users (PUs), and an unlicensed band allocated to non-licensed users, referred to as secondary users (SUs) [6]. According to a report by the US Federal Communications Commission (FCC), a considerable portion of the radio spectrum is underutilized and there are some vacant spaces, referred as white spaces, which may be efficiently utilized by SUs for communication purposes, as the average spectrum utilization rate ranges from 15% to 85% [7]. Cognitive radio (CR) is a technology that is capable of detecting and using white spaces. It has emerged as a remedy ensuring effective utilization of the spectrum without intervening with the operations of PUs [7], [8].

If CR may be integrated with WSNs, it will be able of overcoming numerous challenges typical of conventional WSNs. The new network paradigm thus formed is called cognitive radio sensor networks (CRSNs) [9]. CR has the ability of knowing the spectrum used in the licensed and unlicensed band, and of utilizing its unused portion in an opportunistic manner. PUs are authorized to use the spectrum at any time, while SUs may utilize the spectrum only in the absence of PUs. The activity of PUs is dynamic [10], hence SUs must be alert enough to track the time periods over which the spectrum is allocated to PUs. However, the SUs may communicate with other SUs only when the spectrum is available. Therefore, CRSN imposes a spectrum awareness constraint. By being aware of spectrum utilization, the nodes may avoid communicating using the crowded licensed band, thereby utilizing the radio spectrum in a more efficient manner [9], [10].

Apart from sensing the surrounding environment, CRSN consumes energy in several different ways, such as:

- sensing the radio spectrum to detect PU presence,
- identifying data transmission and thereby routing data packets,
- to performing channel negotiation.

The rest of the paper is structured as follows: in Section 2, related work on clustering in CRSN, along with their potential drawbacks, are presented. In Section 3, the proposed protocol is explained. Performance analysis of the proposed protocol is performed by comparing it with existing clustering protocols, such as DSAC [11] and CogLEACH [12] in Section 4. Finally, Section 5 presents the conclusion.

2. Related Work

In [11], the authors propose a spectrum-aware clustering algorithm referred to as the distributed spectrum-aware clustering (DSAC) algorithm for CRSN. The algorithm performs clustering by sensing the vacant channels, and thus changes in PU activity. DSAC uses group-wise constrained clustering to minimize energy consumption of CRSN nodes. As the number of active PU nodes increases, more spectrum-aware constraints are imposed on the process of clustering. Hence, DSAC offers poor performance in terms of energy consumption while clustering.

In [12], an extension of the LEACH protocol, knows as CogLEACH, suitable for cognitive environment is presented. Each sensor node sends channel state data to the base station. The expected number of clusters is determined by checking the number of nodes required to cover the area, and by verifying the availability of channels at that node. The cluster head (CH) is chosen based on probability value of high availability of channels. Nodes with a high probability value are chosen as CH. Upon formation of clusters, CH nodes generate TDMA schedules and all cluster members start to transmit data within their allotted timeslots. CogLEACH achieves better throughput and lifetime when compared with the traditional LEACH protocol.

In [13], the authors propose an event-driven spectrum-aware clustering algorithm (ESAC) which forms clusters based on event detection. The algorithm achieves energy efficiency by avoiding re-clustering, as the cluster formed is maintained until the end of the event. Clusters are formed by exchanging control messages, thus generating a control overhead. In dynamic networks, such as CRSN, the exchange of control messages causes a delay. Therefore, ESAC suffers from delay in cluster formation.

The energy-aware routing protocol (EAP) referred to in [4] aims at preserving the node's energy by forming clusters. All nodes within a cluster forward the sensed data to CH whose role is to aggregate the collected data and forward them to the BS. As CH consumes more energy than other nodes due to receiving the sensed information, aggregating data and forwarding them to the BS, the role of CH is assumed, on a rotational basis, by all member nodes. EAP minimizes energy consumption by forming clusters with cluster members joining the nearest CH.

The equalized cluster head election routing protocol (ECHERP) is proposed in [5]. It achieves energy efficiency through balanced clustering of sensor nodes. Using the Gaussian elimination algorithm, ECHERP identifies the total amount of energy consumed and thereby selects CHs needed for cluster formation, so that the overall network lifetime is increased. The node which minimizes the total energy consumption within the cluster is elected as CH in each CH selection round. TDMA slots are created by BS and are broadcast to all nodes by which data packets are transferred between the sensor nodes.

A similar energy-aware routing algorithm (ERA), which is a cluster-based routing protocol, is proposed in [3]. In this protocol, CH is selected based on the residual energy of nodes and on intra-cluster distance. Since CHs have multiple tasks to perform, their energy may be depleted rather easily. Also, if the sensor nodes are not deployed properly in the network, then the distance between CHs may vary, resulting in an increase of intra-cluster distance. This results in quicker energy depletion for a CH. The algorithm constructs a directed graph of CHs as a virtual backbone to aid data routing. That is how ERA balances energy consumption between its CHs.

3. Implementation Details

In this section, a novel approach to CRSN that emphasizes energy conservation and forms clusters by assigning idle channels to SUs using binomial distribution is described. Sensor nodes send messages over the available channels. In CRSN, an additional constraint of detecting PU activity is to be considered, as CR nodes may utilize the spectrum only in the absence of PUs. So, based on the results of spectrum sensing and on the availability of channels, CR nodes communicate with each other. The proposed protocol forms clusters setting up communication between the cluster member (CM) and CH, and then CH connects with the neighbor CH, until the BS is reached. The communication process may be divided into two categories: between the clusters (inter-cluster) and within the clusters (intra-cluster). During inter-cluster communication, CH collects and forwards the information to BS through the channels available, which are shared with upstream neighboring clusters. During intra-cluster communication, the sensed information is sent to CH using a local channel.

3.1. Network Model

Let us consider a CR sensor network $P = \{S, N\}$, where $S = \{s_1, \ldots, s_n\}$ indicates the total number of sensor nodes in the network and $N = \{n_1, \ldots, n_{ij}\},\ i \neq j$ indicates connections created between other peer nodes from $N.n_{ij}$ indicates the direct connection between two acquaintance nodes, known as n_i and n_j, within the definite time interval t. All sensor nodes are positioned randomly throughout the grid which examines the updated channels for efficient connection. The network is made up of multiple PUs, SUs with sensing ability, and a centralized BS – as shown in Fig. 1.

Each node maintains a neighbor connectivity table, containing a list of immediate neighbor nodes which are one hop away (Table 1). If the value for node j is set to 1, then the node becomes an immediate neighbor of node i. Sensor nodes with the channel sensing ability are deployed. Sensor nodes are aware of the available channel spectrum,

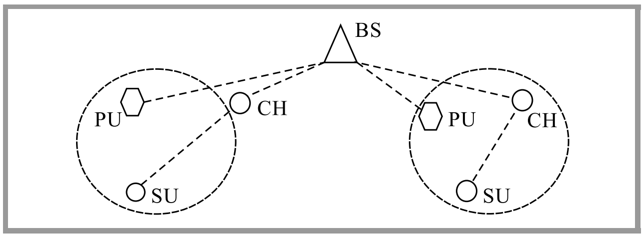

Fig. 1. Secondary user communication via primary channel and CH.

since each sensor node supports cognitive analytical features. This makes the sensor nodes aware of which part of the spectrum is currently underutilized. Each node maintains the transport state timer to update channel state information and channel state probability. The SUs identify channel occupancy and availability during the sensing process according to Table 1.

Table 1
Neighbor connectivity and channel occupancy scheme

Node	A	B	C	D	E	F	G	H	I	Neighbors
A	0	1	0	0	0	0	1	0	0	B, G
B	1	0	1	0	0	0	0	1	0	A, C, H
C	0	1	0	1	0	1	0	0	0	B, D, F
D	0	0	1	0	0	0	1	0	1	C, G, I
E	0	0	0	0	0	1	0	1	0	F, H
F	0	0	1	0	1	0	0	0	1	C, E, I
G	1	0	0	1	0	0	0	1	0	A, D, H
H	0	1	0	0	1	0	1	0	1	B, E, G, I
I	0	0	0	1	0	1	0	1	0	D, F, H

3.2. Channel Allocation Model

The BS requests the sensors to perform channel sensing in order to identify the free channels. The channel list is maintained at BS and is always kept up to date. Each SU performs local sensing to identify its own vacant channels. Since CRSN is a dynamic environment where PU activity is random and PUs may occupy a given channel at any time, channel availability information must be available to the neighboring nodes as well. After performing local sensing, SUs exchange information with their neighbors.

The base station (BS) periodically sends an announcement as a network-wide broadcast message. When all sensor nodes receive it, SUs get initialized with the timer in order to periodically update location- and channel occupancy-related values. All sensor nodes forward their location information to BS which aggregates the collected information and then starts the clustering process. Next, the probability of the node to be a member of a cluster is computed as the ratio between the product of the total number of channels and the collection of idle channels available at the node, and the entire collection of idle channels at all nodes. We use binomial distribution to distribute idle channels sensed by the node, as illustrated in Fig. 2. The grouping and channel allocation process is improved by considering channel overlapping and channel interference computations.

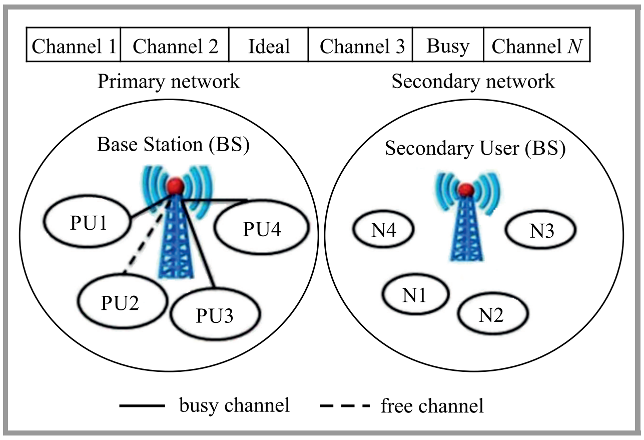

Fig. 2. Distribution of idle channels for secondary users.

3.3. Distance between Nodes and Channel Overlapping

The computation of neighbor distance is based on the position of network region points (X,Y). Let node i and node j have the position of (X_1,Y_1) and (X_2,Y_2), respectively. Distance D_i between node i and node j is:

$$D_t = \sqrt{|X_1 - X_2|^2 \cdot |Y_1 - Y_2|^2} \,. \tag{1}$$

The proposed protocol deals with channel overlapping if two or more SUs share the same channel. To cope with this, ICBCA checks for all available channels in the communication. Based on the derived optimal number of CHs, clustering probability for each node is determined by comparing the result with the total number of cognitive sensor nodes in the network. Channel IDs are updated continuously from the entire network area and the list of channels (C_L) is framed at each sensor node. Let O_L indicate the count of overlapping channels. We then check O_L states from C_L and update the count. If O_L is found in C_L based on network region N_R, the overlapping counts $\pm O_{LC}$ are added and O_L is updated:

$$O_L = \frac{D_{ij} + D_{ti}}{N_R} \,. \tag{2}$$

Then, the final O_L is:

$$O_L = \frac{O_L}{O_{LC}} \,. \tag{3}$$

The channel overlapping state is shown in Fig. 3 and Table 2. If a sensor node falls under the communication range between two clusters, it becomes a one-hop neighbor of two equivalent CHs. This node is referred to as a midway node. Its CH is selected by considering any of the midway sensor nodes as the connection sensor required to reach the next cluster. We refer to the connection sensor node as a one-hop access node. In the other case, if two clusters are non-overlapping but there are sensors falling

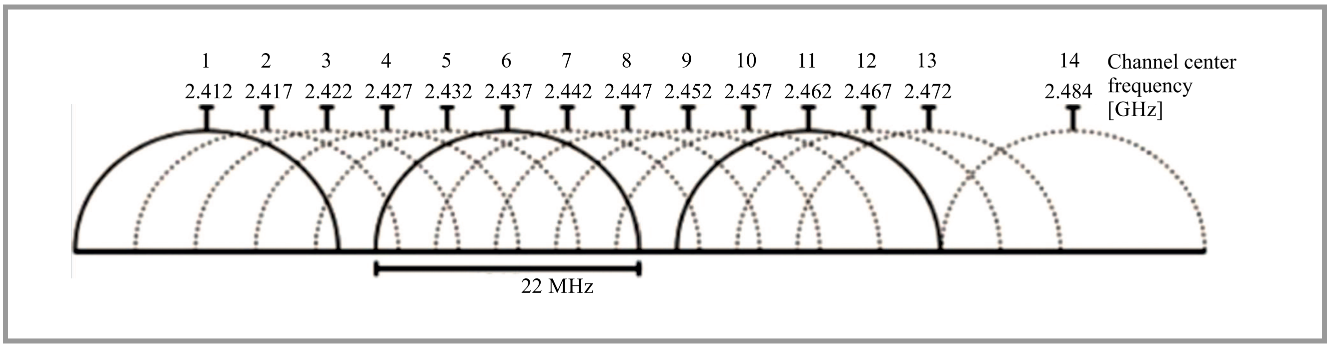

Fig. 3. Channel overlapping illustration.

Table 2
Channel interference and overlapping

Node	Channel 1	Channel 2	Channel 3	Channel 4	Node – channel		Node	Co-channel interference	Overlapping
A	0	1	0	0	A	2	A	–	B–G
B	0	0	0	1	B	4	B	H	A–C
C	0	0	1	0	C	3	C	–	B–D–F
D	1	0	0	0	D	1	D	–	C–G–I
E	0	1	0	0	E	2	E	–	F–H
F	0	0	0	1	F	4	F	–	C–E–I
G	0	0	1	0	G	3	G	H	A–D
H	0	0	1	0	H	3	H	B–G	E–I
I	0	1	0	0	I	2	I	–	D–F–H

within communication range of two clusters which can listen to each other, then these sensors intersect two groups. Those sensors are called two-hop connection sensors.

3.4. Channel Interference

In CRSN, since PU has the highest priority, the active PUs can use all available channels, causing potential interference. Channel interference is considered to be the region where two or more sensor nodes use the same frequency. Channel interference in a specific cluster may be avoided effectively by adopting a policy under which no currently reachable channel within a cluster is selected, as shown in Table 2. By using the binary co-channel conflict list, we divide channel x by sharing it with both i-th and j-th SUs. Table 3 shows the other channel parameters used in the simulation.

Table 3
Channel parameters used in simulations

Parameter	Value
Height of antenna	1.5 m
Packet size	2 MB
Receiver noise level	−2 dBm
Antenna gain	20 dBi
Path loss	4 dBm

3.5. Binary Matrix Formation

Next, the spectrum is divided into several channels, as shown in Fig. 4. The real channel refers to the original spectrum which is available for communication. It divides the frequency band into sub-channels, with the necessary spacing between them. As some channels may be unavailable for use, overlapping may occur, but initially all nodes find a free channel which is represented as the binary matrix in Table 4.

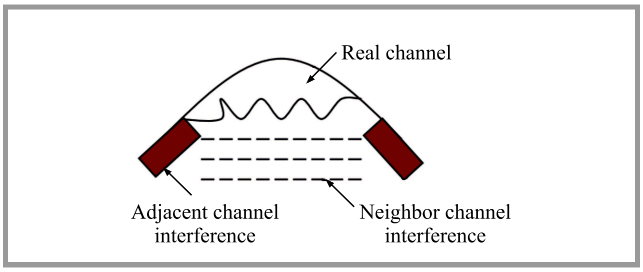

Fig. 4. Sectioning the spectrum into sub-channels.

If the channel is accessible, then the value is set to 1. Otherwise, it is set to 0.

For each potential channel, the incentive cost is computed based on its availability, by checking the distance to PU, throughput and bandwidth. The bandwidth of the communication link determines the packet data arrival rate. The

Table 4
Binary matrix design

Node	Channel 1	Channel 2	Channel 3	Channel 4
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	1	0
5	0	1	0	0

higher the bandwidth value, the higher the data rate, as less time is required for packets to reach their destination. Based on the connection between two sensor nodes, channel bandwidth may be estimated as:

$$C_B = \text{Payload} \cdot \frac{B_W}{C_T} , \tag{4}$$

where C_T and B_W define the current time and bandwidth.
Throughput T_P is defined as the total number of receiving bytes (payload) per second at the receiver end:

$$T_P = \frac{\text{Payload}}{C_T} . \tag{5}$$

If the channel overlapping count is > 0 and the channel interference range is > 0, then the communication state is computed. The occupied number of channels is updated from the channel list C_L. We then find the number of sensor nodes S from the network and update the optimal cluster count K. Next, channel information C_I is found based on the number of channels m and occupied channel list O_C.

$$C_I = m - O_C . \tag{6}$$

Then, the state of the cluster and CH after each interval R of the communication range C_R is computed:

$$CR = \frac{K}{S-K} \frac{S}{K} \cdot R \ [\%] . \tag{7}$$

The C_I value is updated periodically by using channel information from current scenarios $C_{I\_}$:

$$C_{I\_}(m - C_I + 1) . \tag{8}$$

The updated channels using the channel value α are:

$$\alpha = \frac{m}{C} , \quad C_R = \frac{K\alpha C_I}{m, 1} . \tag{9}$$

If the number of updated channels remains unchanged, then:

$$C_{R\_} = \min \frac{KC_I}{\sum C} . \tag{10}$$

The effective channel selection is based on transmission range $C_{R\_}$. This effective channel selection is required to identify the optimal number of available idle channels within the PU coverage area. Next, the probability of a node to become a CH is estimated. The number of sensor nodes within the PU communication range, known as PU-affected area, with the coverage probability and the number of sensor nodes not covered within PU, is computed from the affected area with the non-coverage probability. These factors are used to evaluate the final probability of the node to become the CH, and then to identify the transmission channel TCcount and the receiving channel RXcount.
During communication, static channel probability C_P for two sensors is:

$$s1 = sC_P \quad \text{and} \quad s2 = s(1 - C_P) . \tag{11}$$

If $C_P > 0$ and $C_P < 1$, channel selection C_S is:

$$C_S = \min \frac{KC_I}{\left[(s1m + s2m), 1\right]} . \tag{12}$$

Further, an available channel C_A is defined as 1 if the channel is vacant and 0 otherwise. The channels are chosen from Table 2, as available channel C_A and the value of $(C_A C_S)$ are computed. In the event of channel overlapping and when the interference range is greater than 0, the channel incentive cost is derived and assigned to each sensor node using:

$$\text{cost} = \frac{C_A \cdot C_S}{(O_I + C_I)} . \tag{13}$$

3.6. Avoiding Channel Interferences

Interference between channels occurs when two sensor nodes use the same channel at the same time. Here, we avoid co-channel interference by restricting channel reuse within a single cluster. Channels are assigned to clusters ensuring that the same channel cannot be reused during the specific iteration. Channel nomination is performed based on priority-based allocation. Based on current channel ownership and potential of interference with the neighboring node's channels, the process of clustering sensor nodes is performed. The highest clustering probability of the node based on channel ownership and the lowest channel interference value is estimated. The clustering of sensor nodes is performed among all cognitive sensor nodes in the network. Sensor nodes collect and update information about their neighbors to determine the optimum quantity of clusters needed to cover the entire network area. Then, node density is evaluated as the total number of nodes per unit area. The average cluster distance is estimated by calculating the range from the present CH to all other cluster members in the cluster by checking whether a node is a CH or not.
The largest distance value is measured by taking the highest range value of the present CH to the corresponding cluster group members, as shown in Fig. 5. From these values, we estimate the average power needed to perform data transmission, and cost D_t.
The inter-cluster communication power rate is computed as $I_N = kC_O R_P 2D_t$ and intra-cluster communication power rate is computed as $I_G = 2C_O R_P \sum D_t$, where $k = \frac{S}{D2D_t}$, $D = \frac{S}{N_S}$, and N_S denotes the network size. The total communica-

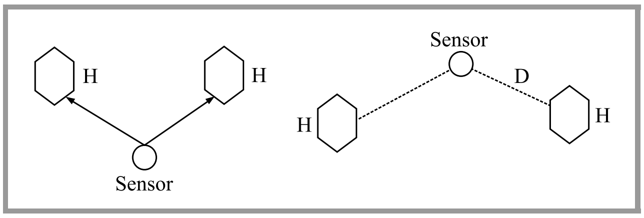

Fig. 5. Sensor and CH communication.

tion power necessary to perform data transmission in inter-cluster I_N and intra-cluster I_G is:

$$Total\ power = 2C_O R_P E_E + 2E_E K C_O R_P 2D_t \ . \tag{14}$$

3.7. CH Election and Clustering Process

Once the optimal collection of tentative clusters is identified, the lowest power expected to complete the communication process is computed. Then, the probability estimation-based stable CH is selected by using bandwidth, throughput, channel value and distance to SU. Based on this, the node executes the CH decision process to join a particular cluster. The node with the highest probability value becomes the tentative CH.

The selected CH nodes activate the final CH timer to receive a join message from the group members. Upon hearing the tentative CH announcement, non-CH nodes store the ID, location and the probability value of the CH. The non-CH nodes start the wait timer to receive an announcement message from all CH nodes within their coverage. Once the wait time expires, non-CH nodes select the best available CH with the highest probability and lowest distance. The CH nodes keep the member list by accepting the join information from all the members. The CH validates the member list for the joined nodes. If the non-CH node do not receive the final CH announcement message, then it tries to join other potential groups. If there are no available CH within range, the node itself becomes a CH.

The node with the highest channel availability rate and the lowest level of interference will become the CH. In certain iterations, the selected nodes should not be reassigned with the same channel until all channels are reassigned based on the round-robin method.

3.8. Selection of Fair Channel and Slot Computation

The fairness index is computed by calculating the number of channel allocations assigned to each sensor node during the current iteration. The fair value is calculated by applying the fairness index with the maximum incentive cost and channel communication parameters. For each node, the required data rate is computed by checking all incoming and outgoing traffic. By comparing the required data rate of each node, ICBCA estimates the priority of the data transmission in the node.

During the channel allocation process for each cluster, the channel with the non-overlapping and the minimum interference gets a high fairness index value and a high data priority rate. The same process is repeated for all formed clusters and then data transmission takes place in the allocated channel. Upon completion of cluster formation, the CH node generates TDMA slots for all its cluster members. The generated TDMA schedules are broadcasted to all member nodes within the cluster. Based on the received TDMA slot, the SUs perform the data transmission operation.

4. Performance Evaluation

In this paper, Network Simulator NS-2 was used to simulate the proposed ICBCA protocol and to compare the results with existing clustering algorithms, i.e. DSAC [11] and CogLEACH [12].

Figures 6 and 7 show the comparison of protocols in terms of energy consumed by the node and in terms of the node's residual energy. CogLEACH considers the probability of finding idle channels and the node which has more idle

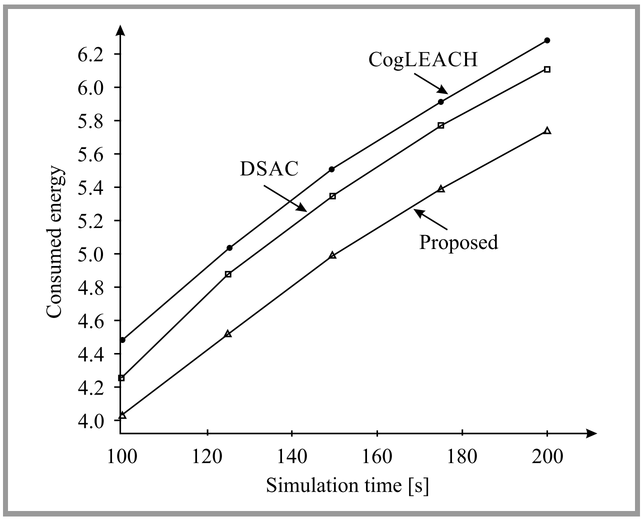

Fig. 6. Simulation time vs. average energy consumed by the node.

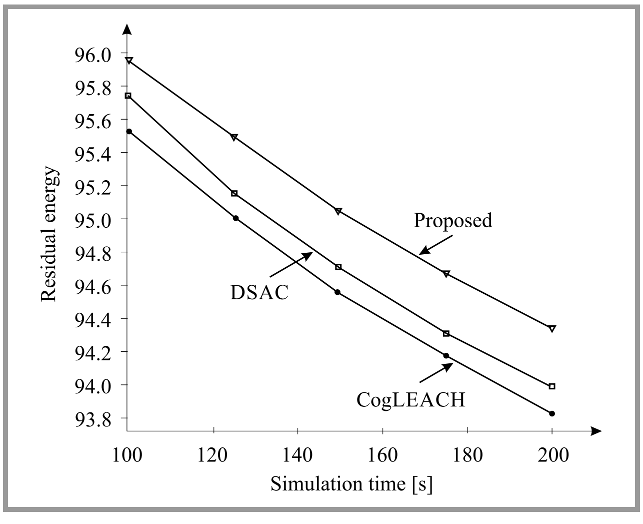

Fig. 7. Simulation time vs. average residual energy of the node.

channels becomes the CH. It prohibits SUs from transferring packets if the entire spectrum is busy, thereby saving energy. However, in a dynamic environment like CRSN, this approach is not suitable. In DSAC, energy is saved based on the CH rotation policy. It considers the optimal number of clusters by avoiding overlapped channels. ICBCA offers better energy efficiency than other solutions, since both channel interference and channel overlapping are taken into consideration. In terms of residual energy, ICBCA has better efficiency too when compared with CogLEACH [12] and DSAC [11].

The throughput of CogLEACH and DSAC is comparatively lower than that of ICBCA, as both of them do not support separate channels for transferring control messages (Fig. 8). Although DSAC relies on channel overlapping and interference to select the best channel, priority and fairness-based channel selection are not used. ICBCA, in turn, uses separate channels for control messages and for data transmission. It also performs priority-based channel selection, which results in a better throughput.

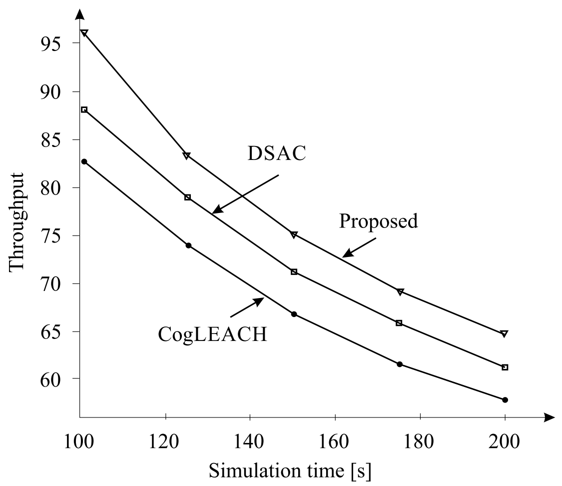

Fig. 8. Simulation time vs. throughput.

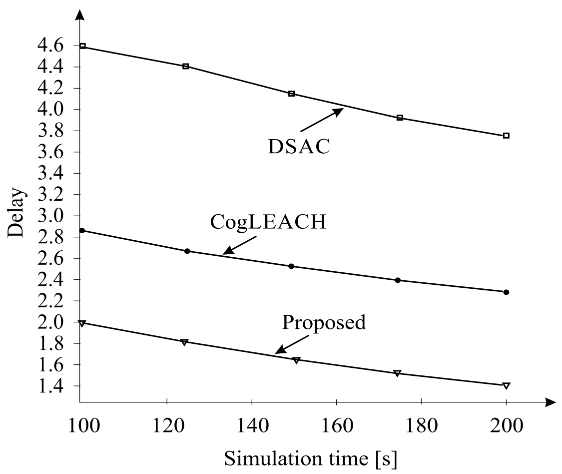

Fig. 9. Simulation time vs. end-to-end packet delay.

Figure 9 compares packet delay between the protocols. As mentioned earlier, ICBCA has a separate channel for sending data packets and for exchanging control messages, due to which the destination nodes suffer from a lower delay. Since ICBCA implements cluster-based channel allocation, allocating a dedicated channel only after the cluster is formed, the proposed work offers a better packet delivery ratio compared with DSAC and CogLEACH (Fig. 10).

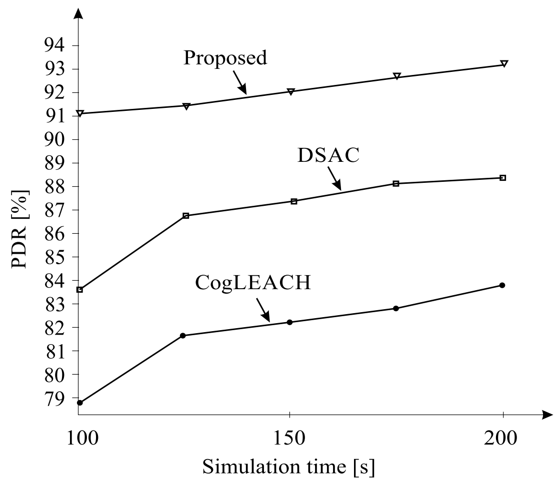

Fig. 10. Simulation time vs. packet delivery ratio.

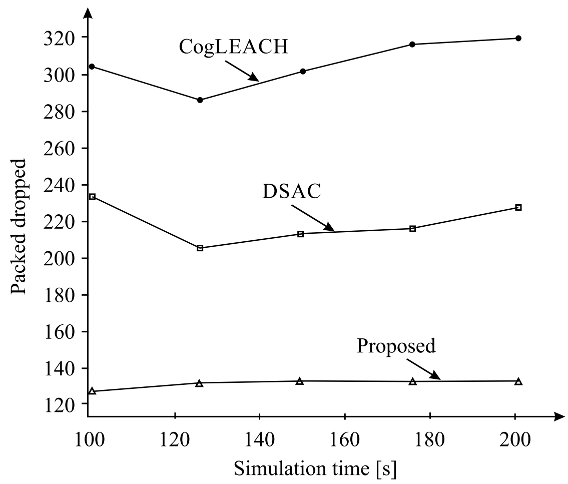

Fig. 11. Simulation time vs. packets dropped.

In ICBCA, packet drop is lower when compared with DSAC and CogLEACH, as both these protocols do not consider priority and fairness-based allocation of the dedicated channel. Figure 11 shows the comparison of the numbers of packets dropped.

5. Conclusion

In this paper, a novel clustering algorithm is implemented, relying on fairness- and priority-based allocation of channels to clusters. According to our knowledge, the existing literature does not describe any research in which a channel assignment scheme considering data rates and quality-of-service (QoS) parameters would be relied upon. In this area, ICBCA shows a significant improvement in the network lifetime of nodes. Simulation results show that ICBCA offers better performance in terms of throughput, energy consumption, packet delay and packet delivery ratio when compared with other existing CRSN clustering algorithms, such as DSAC and CogLEACH.

References

[1] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey", *IEEE Wirel. Commun.*, vol. 11, no. 6, pp. 6–28, 2004 (DOI: 10.1109/MWC.2004.1368893).

[2] C. Buratti *et al.*, "An overview on wireless sensor networks technology and evolution", *Sensors*, vol. 9, no. 9, pp. 6869–689, 2009 (DOI: 10.3390/s90906869).

[3] T. Amgoth and P. K. Jana, "Energy-aware routing algorithm for wireless sensor networks", *Comp. & Elec. Engin.*, vol. 41, pp. 357–367, 2015 (10.1016/j.compeleceng.2014.07.010).

[4] M. Liu, J. Cao, G. Chen, and X. Wang, "An energy-aware routing protocol in wireless sensor networks", *Sensors*, vol. 9, no. 1, pp. 445–462, 2009 (DOI: 10.3390/s90100445).

[5] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, "Energy efficient routing in wireless sensor networks through balanced clustering", *Algorithms*, vol. 6, no. 1, pp. 29–42, 2013 (DOI: 10.3390/a6010029).

[6] B. Wang and K. J. Ray Liu, "Advances in cognitive radio networks: A survey", *IEEE J. of Selec. Topics in Sig. Process.*, vol. 5, no. 1, pp. 5–23, 2010 (DOI: 10.1109/JSTSP.2010.2093210).

[7] M. Cesana, F. Cuomo, and E. Ekici, "Routing in cognitive radio networks: Challenges and solutions", *Ad Hoc Netw.*, vol. 9, no. 3, pp. 228–248, 2011 (DOI: 10.1016/j.adhoc.2010.06.009).

[8] I. Pefkianakis, S. H. Y. Wong, and S. Lu, "SAMER: Spectrum aware mesh routing in cognitive radio networks", in *Proc. 3rd IEEE Symp. on New Front. in Dynam. Spec. Access Netw.*, Chicago, IL, USA, 2008 (DOI: 10.1109/DYSPAN.2008.90).

[9] G. P. Joshi, S. Y. Nam, and S. W. Kim, "Cognitive radio wireless sensor networks: applications, challenges and research trends", *Sensors*, vol. 13, no. 9, pp. 11196–11228, 2013 (DOI: 10.3390/s130911196).

[10] H. Kim and K. G. Shin, "Adaptive MAC-layer sensing of spectrum availability in cognitive radio networks", Tech. Rep. CSE-TR-518-06, University of Michigan, 2006 [Online]. Available: https://www.eecs.umich.edu/techreports/cse/2006/CSE-TR-518-06.pdf

[11] H. Zhang *et al.*, "Distributed spectrum-aware clustering in cognitive radio sensor networks", in *Proc. IEEE Global Telecommun. Conf. GLOBECOM 2011*, Houston, TX, USA, 2011 (DOI: 10.1109/GLOCOM.2011.6134296).

[12] R. M. Eletreby, H. M. Elsayed, and M. M. Khairy, "CogLEACH: A spectrum aware clustering protocol for cognitive radio sensor networks", in *Proc. 9th Int. Conf. on Cogn. Radio Oriented Wirel. Netw. and Commun. CROWNCOM 2014*, Oulu, Finland, 2014, pp. 179–184 (DOI: 10.4108/icst.crowncom.2014.255370).

[13] M. Ozger and O. B. Akan, "Event-driven spectrum-aware clustering in cognitive radio sensor networks," in *Proc. IEEE INFOCOM 2013*, Turin, Italy, 2013, pp. 1483–1491 (DOI: 10.1109/INFCOM.2013.6566943).

Veeranna Gatate has completed his B.Eng. in Computer Science and Engineering from the Appa Institute of Engineering and Technology, Kalaburagi, Karnataka, India, in 2007 and post graduation from the Poojya Doddappa Appa College of Engineering from Kalaburagi, Karnataka, India, in the year 2012 under affiliation of Visvesvaraya Technological University (VTU) Belagavi, Karnataka, India. His research areas include wireless sensor networks, body area networks, mobile ad-hoc networks. He is currently working as an Assistant Professor in Appa Institute of Engineering and Technology Kalaburagi, Karnataka, India.

https://orcid.org/0000-0003-3189-0870
E-mail: vcgatate@gmail.com
Computer Science and Engineering Department
Poojya Doddappa Appa College of Engineering
Kalaburagi, Karnataka, India

Jayashree Agarkhed is currently working as Professor in the CSE Department at PDACE Kalaburagi, affiliated to VTU. She obtained her M.Tech. in CSE in 2003 and Ph.D. in 2013 from VTU. Her main research areas are in wireless sensor network, multimedia information networks, artificial intelligence, cloud computing, and Internet of Things. She has published more than 150 scientific articles in top-tier journals and conferences and book chapters. She is the an author of 2 books in the field of computer science.

https://orcid.org/0000-0003-3365-6498
E-mail: jayashreeptl@yahoo.com
Computer Science and Engineering Department
Poojya Doddappa Appa College of Engineering
Kalaburagi, Karnataka, India