Decision-Tree Based Transceiver Selection for Medium Access Control in Wireless Sensor Networks

Arnold Chau
HME Clear-Com Ltd
Cambridge, United Kingdom
arnold.chau@clearcom.com

John F. Dawson
Dept. of Electronic Engineering
University of York
York, United Kingdom
john.dawson@york.ac.uk

Paul D. Mitchell
Dept. of Electronic Engineering
University of York
York, United Kingdom
paul.mitchell@york.ac.uk

***Abstract* — The primary contribution of this paper lies in evaluating the potential benefits of using decision tree based transceiver selection algorithms for multi-hub wireless sensor networks (WSNs) to improve network performance and stability. The classification algorithm helps improve network throughput and reduce unnecessary transceiver handovers by considering multiple decision factors, without any additional hardware complexity at the sensor nodes. The algorithms allow the nodes to use information included in the reference messages from the transceivers, and the Receiver Signal Strength Indicator (RSSI) as decision factors. This paper demonstrates the effectiveness of decision tree based algorithms (Decision Tree and Random Forest) through simulation, and the results show that this approach can enhance throughput and provide extra stability to the WSN in the scenarios considered, compared to traditional approach using distance or RSSI.**

Keywords— Decision Tree, Random Forest, Wireless Sensor Network, Transceiver Selection.

I. Introduction

Wireless sensor networks (WSNs) have been employed in a wide range of health care, industrial and environmental monitoring applications [1-3]. In some cases, node mobility is required. In order to provide coverage for a large deployment area, or to improve the network performance, some WSNs use hubs with multiple transceivers. There are a number of Medium Access Control protocols that employ multiple transceivers [4-8]. However, consideration of handover for mobile nodes between hubs has been very limited. Mobile nodes in a multi-hub network introduce instability due to handovers between hubs, impacting the performance of the application [4].

Decision Tree [9] is a popular supervised machine learning classification algorithm. Decision Tree algorithms have been used in wide range of applications in wireless communication such as in routing and cluster head selection [10]. Using a decision tree, different decision factors and possible options can be laid out in an effective manner. This also gives us a balanced view of risks and rewards associated with each possible outcome. A random forest approach is simply a collection of randomly initialized decision trees with different combinations of decision features, in which the results are aggregated into one final decision [11]. This approach reduces common issues in machine learning such as overfitting and bias. Both classifier methods are applied on the dataset containing information of the decision features, using Python Scikit-Learn library package [12], a python framework widely used by researchers for machine learning algorithm implementation. In the simulated scenarios, the same set of performance metrics are used to measure and evaluate the predictive ability of both classification models. Metrics including accuracy, precision, recall, and F1 score, are widely used in the literature for classification algorithms [13].

In this paper we present an implementation of a decision tree based classification algorithm to solve the transceiver selection problem. In Section II the model is described, including the considerations of the scenarios, assumptions and decision factors. Section III presents the performance evaluation, with our conclusions presented in Section IV.

II. Model Description

A. WSN Scenarios

In WSNs, it is common to add/remove nodes to an existing network. In a network with multiple hubs, each hub equipped with multiple transceivers, the decision of which transceiver to connect to becomes an important consideration. Consider a large network with nodes dispersed in an area, where each node belongs to a particular cluster as shown in Fig 1. Each cluster has one hub which connects to all nodes within the cluster. All nodes will send their data to the connected hub through one of its transceivers. All the hubs are assumed to be connected to the same sink, and all transceivers at each hub operate on a single frequency channel. Adding a new node to a cluster can affect the performance of the network. With a contention based access scheme the more nodes there are in a cluster competing for access, the higher the probability of packet collisions. The goal of this proposed method is to consider the effect of different decision metrics other than simply considering a Receiver Signal Strength Indicator (RSSI) and distance, when choosing which hub a node should connect to.

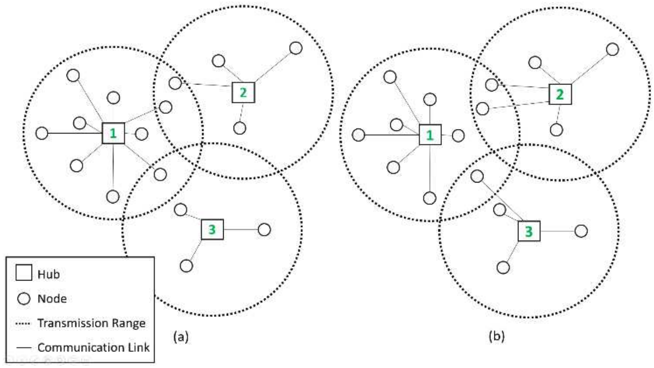

Figure 1. The WSN topology showing clusters of nodes with each connected to a single hub.

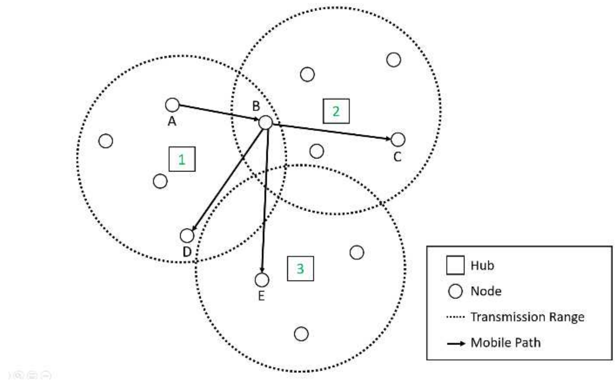

Figure 2. Example movement path of sensor node in mobile scenarios.

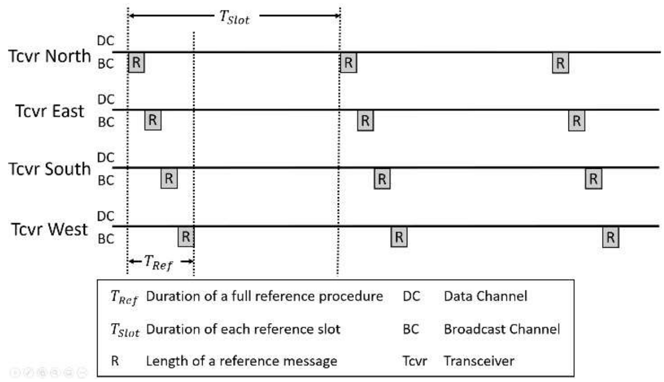

Figure 3. Frame structure of transceiver reference message.

In some applications a node might manoeuvre within the vicinity of a set of hubs. Instead of handing over between hubs due to the shorter distance or higher RSSI/ Signal-to-Interference-and-Noise Ratio (SINR), in some cases it might be beneficial for the device to stay connected to a single hub to avoid unnecessary handovers. In wireless or cellular networks, handovers are often decided by the distance between the transceiver and the device; or the measured RSSI/SINR levels. If the distance to a transceiver becomes smaller than the distance to the hub which a node is currently connected, or if the RSSI or SINR for the new transceiver is higher than a pre-determined threshold, the device will handover to the new transceiver. However, the threshold for the minimum RSSI/SINR highly depends on the environment, the propagation characteristics can vary significantly in space and time due to shadowing, and a single threshold value might not be feasible for a dynamic network. Instead of using a fixed value as the threshold to determine the transceiver selection, the proposed approach using classification algorithms provides a solution that can be adaptive to different environment and applications.

In this paper, we define unnecessary handover as one that occurs when adequate communication could be maintained (and has been provided) without handover. For example, in Fig 2, a node is moving from A to B. At position B, using traditional methods based on distance or RSSI/SINR, the node will handover to transceivers on hub 2, while still within the transmission range of hub 1. In the case of the node staying in position B, or moving towards position C, this might be a good decision. However, if the node then moves towards position D, it will have to handover to transceivers on hub 1 again; or if the node moves towards position E, it will have to be handover to transceivers on hub 3. This causes unnecessary node handovers, decreasing the performance of the network.

Although traditional transceiver selection techniques such as localization using Global Positioning System (GPS) or K-means clustering might be useful for outdoor applications, transceiver selection using position detection clustering might not be suitable for indoor applications or more dynamic environments. Many of these techniques assume there is a direct relationship between the distance of the transceiver and the RSSI or SINR measure, and that as long as the RSSI or SINR is above a certain threshold, a connection can be established. However, these techniques assume that the predetermined RSSI or SINR value is suitable for all environments, without providing consideration of the dynamic environments and sensor node behavior. They also fail to consider other decision metrics such as node mobility and node density. Therefore, there is a great need for a dynamic transceiver selection mechanism that considers more relevant factors.

B. *Model Assumptions*

In this paper, we consider n nodes randomly distributed in an area with i hubs, where each node is connected to a hub, so that clusters are formed. The information regarding each cluster such as node density, packet loss ratio, and throughput, are included in the reference message broadcast by its transceivers using the broadcast channel. The formats of the frames for the transceiver reference message are shown in Fig 3. The transceivers take turns to send reference messages within the fixed reference frame duration T_{Ref}, where time is synchronized and divided into slots (T_{Slot}) with a duration of 10 ms.

It is assumed that the DIFS-VSDH Medium Access Control (MAC) protocol is used by the WSN [5] as a useful example. Each hub is equipped with four transceivers pointing East, South, West, and North. Each transceiver is equipped with one antenna for data transmission and an additional antenna, with the same antenna pattern, for broadcasting reference messages. The behaviour of the hub will follow the behaviour of the DIFS-VSDH protocol, with the addition of broadcasting reference messages periodically. The reference messages are broadcasted in turns, with the North antenna at each hub first to broadcast, then East, South, and West in turns. No additional hardware is required at the nodes to minimise the energy consumption and manufacturing cost. However, the node protocol is a little more complex, as the node needs to decide when to switch to the broadcast channel to receive reference messages for the purpose of transceiver selection. Reference message from all visible transceivers will be used by the classification algorithms at the nodes. The reference message is also used by the nodes to calculate the RSSI. Nodes will switch to the broadcast channel periodically (every T_{slot}) to receive the reference messages. As illustrated in Fig 4, sensor nodes can decide not to switch to the broadcast channel if there is an ongoing transmission. As the reference messages are broadcast periodically and frequently, the sensor node will wait for the next slot to receive a reference message. If the sensor node is in the backoff state, based on network allocation vector (NAV), it will switch to the broadcast channel to receive the reference message without pausing the countdown timer. However, if the remaining time on the countdown timer is not sufficient for the node to receive the

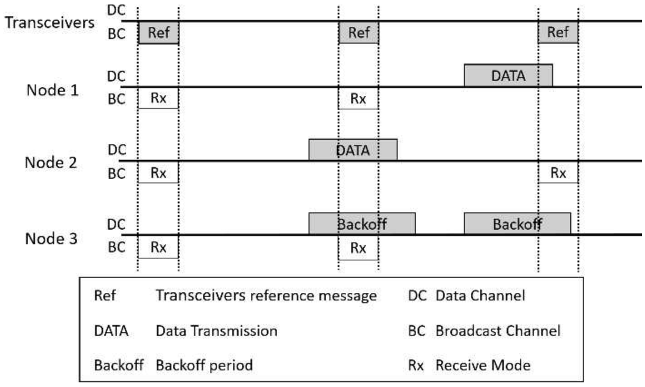

Figure 4. Examples of broadcast and data channel frame structure.

reference message, the sensor node will not switch to the broadcast channel, and it will wait for the next broadcast slot.

As we assume nodes may move, as may obstacles and sources of interference, periodically reviewing the transceiver selection allows sensor nodes to connect to the appropriate transceiver dynamically. For simplicity but without losing generality, it is assumed that all nodes are within the transmission range of at least one transceiver when they are active. Although all nodes are within the transmission range of at least one transceiver, they are likely to be within the transmission range of multiple transceivers. Hence, the need for a transceiver selection.

C. *Decision Factors*

The proposed approach is based on the following factors:

1. **Receiver Signal Strength Indicator (RSSI):** Each node calculates the RSSI of each transceiver using the reference message on the broadcast channel. The higher the RSSI, the higher the probability of connecting to this transceiver.
2. **Packet loss ratio:** The packet loss ratio of transceiver will be included in the reference message broadcasted by the transceivers periodically. Nodes might be connected to different transceivers over time, and this factor tells us which transceiver might be able to provide a reliable communication link. The lower the ratio, the higher the probability of connecting to this transceiver.
3. **Node density at each transceiver:** The number of nodes connected to each transceiver will be included in the reference message broadcasted by the transceivers periodically. A high node density leads to a high packet collision probability and longer delay due to backoff and retransmissions. The lower the value, the higher the probability of connecting to this transceiver.
4. **Node Mobility behaviour:** The mobility of the node has a great impact on the stability and performance of the network. The topology of the network may be changed very frequently due to mobile nodes. The analysis of the mobility of the nodes can help decide if the node should connect to a new transceiver, or if it should remain in its current cluster. Information from previous reference messages is used to calculate the rate of change in RSSI and SINR to be used as the decision factor. This allows sensor nodes to decide which transceivers are likely to provide constant adequate connection with minimal handovers.
5. **Time connected to each transceiver:** In order to reduce unnecessary handovers, we consider the time duration of connection between the node and each transceiver. The longer the connection time, the higher the probability of connecting to this transceiver.

As each transceiver can receive packets from all nodes within transmission range, simultaneous transmission from multiple devices to the same transceiver will result in packet collisions. Therefore, maintaining the balance between having sufficient signal strength and the number of devices connecting to each transceiver is essential. The goal of the proposed approach is to exploit the network capacity efficiently, avoiding traffic congestion resulting from high node density at some transceivers, and reducing the number of unnecessary device handovers between transceivers to improve stability.

The proposed algorithms can be used in different WSN scenarios, such as 1) when a new device is added to the network and connects to a transceiver based on the received reference message; 2) a connected node periodically checking and deciding if it should handover to another transceiver based on the received reference message.

III. PERFORMANCE EVALUATION

In this section we compare the proposed decision tree and random forest classification algorithms by implementing 10 randomly generated WSN topologies using the Riverbed Modeler [13]. Nodes within the transmission range of multiple transceivers will use one of the algorithms to decide which transceiver to connect to. The simulation parameters are displayed in Table 1. The DIFS-VSHC MAC protocol [5] is used as an example to demonstrate the effectiveness of the transceiver selection algorithms. The sensor nodes are deployed at random coordinates using a pseudorandom number generator with a random distribution. The trajectories of the mobile nodes are defined by generating coordinates using the pseudorandom number generator. The velocity of each node is randomly generated using pseudorandom number generator, with a uniform distribution between 1 to 5 miles per hour.

TABLE 1. SIMULATION PARAMETERS

Parameters	Values
MAC Protocol	DIFS-VSDH-MAC
RTS, CTS, ACK length	32 bits
Channel bit rate	250 kbit/s
Data packet length	2048 bits
Number of Hubs (i)	3
Decision Features	9
Transceivers per hub	4
Data Channels	3
Broadcast Channel	1
Reference Message length (R)	32 bits
Reference Message Slot (T_{Slot})	30 ms
Number of Nodes (n)	100, 150
Node Velocity	1-5 mile/hour

A. *Dataset*

Data is collected from all sensor nodes. A set of formatted examples of data, obtained from different WSNs is shown in Table 2 as indicators. A total of over 1000 datapoints were considered in this paper. Each datapoint is a discrete unit of information for making a prediction, where a dataset is a collection of datapoints. The correct labels are provided for all datapoints for evaluating the prediction performance of the classification models. All datapoints are standardized as they are collected from different systems with different topologies. Standardization is done by removing the mean value of each feature from the dataset, and scaling if to unit variance by dividing the features by their standard deviation. A common approach to evaluate the performance of decision tree based models is by splitting the dataset into a training set and a validation set.

In this paper, we compare the K-fold cross-validation method and the hold-out method for dividing the dataset. A K-fold cross-validation is a technique used in data mining to estimate the model performance by splitting the dataset into K-fold (parts), in which K-1 folds will be chosen randomly as training set and 1 fold will be used as test set. 10-fold is a common approach to K-fold cross-validation, however, in general K remains an unfixed parameter. The hold-out method is a splitting method that pre-determines the proportion of training set and test set at the beginning. The common split ratios between training set and test set are 70:30, 80:20, and 90:10. The performance of two algorithms, namely the Decision Tree and Random Forest are evaluated later in the paper.

There are two possible labels, and each dataset can belong to only one of them:

Label I - Do not handover to new transceiver
Label II - Handover to new transceiver

The decision features are labelled as:

- **R** The comparison of RSSI between potential and current transceiver
- **N1** Node density at potential transceiver
- **N2** Node density at current transceiver
- **L1** Packet loss ratio at potential transceiver
- **L2** Packet loss ratio at current transceiver
- **X1** Continuous rate of increase in RSSI with potential transceiver
- **X2** Continuous rate of decrease in RSSI with current transceiver
- **T1** Time connected to potential transceiver from previously statistics
- **T2** Time connected to current transceiver from previously statistics

TABLE 2. FIRST FIVE EXAMPLES OF THE DATASET

Dataset	1	2	3	4	5
R	Lower	Higher	Higher	Higher	Lower
N1	High	Medium	Low	High	Medium
N2	Medium	Medium	Low	Low	High
L1	High	Medium	High	Low	Low
L2	High	High	Medium	Medium	Low
X1	No	No	Yes	Yes	Yes
X2	Yes	No	Yes	Yes	Yes
T1	Short	Short	Medium	Long	Short
T2	Long	Long	Medium	Short	Long
Label	I	II	II	II	II

B. Classification Performance

For the evaluation of the correctness of the results yielded by the proposed models, the following metrics are taken into consideration: Accuracy, Precision Recall, and F1 score. All four classification metrics considered in this paper have a best value of 1, and a worst value of 0. The confusion matrix is shown in Table 3, which is used to investigate the performance of a classification model using actual known test labels. The performance of the classification methods is evaluated in terms of:

- **True Positives (TP):** The number of precisely identified class instances.
- **True Negatives (TN):** The number of precisely identified irrelevant instances to the class.
- **False Positives (FP):** The number of wrongly allocated instances to the class.
- **False Negatives (FN):** The number of undetected class instances.

TABLE 3. CONFUSION MATRIX FOR CLASSIFICATION ALGORITHMS.

	Actual Label	
	Positive	Negative
Positive Prediction	True Positive	False Negative
Negative Prediction	False Positive	True Negative

1) Accuracy

Accuracy is one of the most important evaluation metrics used by classification learning algorithm. In this paper, the accuracy measurement is defined as the proportion of the number of correct predictions over total number of predictions as shown in Eqn. 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2) Precision

While accuracy shows the ability of the algorithm to predict the correct outcome, precision is used to show the ratio of data categorised in each category (label) that actually belongs to that category. In this paper, precision is the ratio of correctly predicting handovers as shown in Eqn. 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3) Recall

Recall on the other hand shows the ability of the algorithm to recognise each category correctly. In this paper, recall is the ratio of the correctly predicted handovers over the sum of correctly predicted handovers and incorrectly negative predictions as shown in Eqn 3.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4) F1 Score

The F1 score is weighted between Precision and Recall. Therefore, this metric takes all four components into account as shown in Eqn. 4. Whilst F1 score is not as easy to understand as accuracy, it is a useful metric especially of there is an uneven class distribution. This is usually used to compare the performance of two classifiers. While solving real-life problem using classification, imbalanced class distribution usually exists. Different from the F1 score, accuracy does not take data distribution into account. This can lead to an incorrect conclusion of which classifier is better for a particular application. For example, if there is a dataset with 100 datapoints, where 90% of them are negative sample and only 10% of them are positive samples. If a classifier predicts all samples to be negative, it will have an accuracy of 90%. However, it will have a very low F1 score.

$$F1\ score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \qquad (4)$$

The aggregated performance of the decision tree algorithm is at 82.7%, $p < 0.019$, whereas the aggregated performance of the random forest algorithm is at 87.5%, $p < 0.03$. Details of the results obtained have been listed in Table 4 and Table 5. The results in the tables represents the aggregated results of different splitting methods by running the classification models 10 times, each time randomising the datapoints within the dataset.

From the data shown in Table 4 and Table 5, we can see that both of the classification algorithms provide a high accuracy of over 80%. However, Random Forest provides a better F1 score. This means using the Random Forest approach, there will be fewer false positive and false negative predictions.

TABLE 4. PERFORMANCE OF DECISION TREE CLASSIFIER.

Splitting Ratio	Accuracy	Precision	Recall	F1
10-fold	82.2%	83.9%	82.3%	83.1%
70 : 30	85.2%	88.9%	72.7%	80.0%
80 : 20	85.7%	83.3%	83.3%	83.3%
90 : 10	80.0%	66.7%	66.7%	66.7%
Average	83.3%	80.7%	76.3%	78.3%

TABLE 5. PERFORMANCE OF RANDOM FOREST CLASSIFIER.

Splitting Ratio	Accuracy	Precision	Recall	F1
10-fold	83.2%	84.8%	79.1%	81.9%
70 : 30	88.9%	81.8%	80%	80.9%
80 : 20	88.9%	83.3%	83.3%	83.3%
90 : 10	90.0%	100%	80%	88.9%
Average	87.8%	87.5%	80.6%	83.8%

C. Impact on WSN Performance

The decision tree and random forest model both have a high accuracy with different splitting ratios in the control environment/dataset. In this section, instead of splitting the dataset, all datapoints are used to train the classification models, and being tested in different WSNs using the simulation tool in order to evaluate the impact on WSN performance.

1) Unnessesary handovers

Handovers introduce instability to the network as stated earlier in Section I. In the mobile scenarios considered in this paper, unnecessary handovers are reduced by 23% and 25% respectively compared to simply considering distance and RSSI/SINR, using the decision tree algorithm and the random forest algorithm.

2) Throughput

Table 6 shows the average throughput improvement using the proposed transceiver selection algorithms compared to the saturated throughput performance of the original DIFS-VSDH-MAC protocol with the same simulation parameters.

It can be seen that the WSN can achieve a higher throughput because of the use of the proposed transceiver selection algorithms, with a slight increase of energy consumption due to reception of the reference messages. To quantitatively compare the energy consumption, quote values from MICAz mote [15] are adopted. Using the proposed algorithm, the additional energy required for sensor node is only increased by approximately 7.9%. This is because use of the classification algorithms improves the distribution of the nodes among clusters, reducing the probability of packet collision due to fewer competing nodes; whilst the reduction in unnecessary handovers reduces packet loss among mobile nodes.

TABLE 6. THROUGHPUT IMPROVEMENTS IN SIMULATED SCENARIOS.

Number of nodes	Decision Tree	Random Forest
100 nodes	15.4%	16.9%
150 nodes	22.0%	26.1%

IV. CONCLUSION

In this paper, we have proposed a simple and dynamic transceiver selection algorithm for WSNs. We have exploited a selection of 9 features from 5 decision factors on the selection of transceivers for nodes. After receiving reference messages at each reference slot, the sensor node runs the decision tree based algorithms and selects the transceiver suitable for connection. Simulation results show that the proposed scheme can improve the network performance and stability without additional complexity at the sensor nodes. Moreover, the accuracy of the decision tree based algorithms depends mostly on the training dataset. The more training data is collected over time, the more accurate the decision tree based algorithms. For future development, we plan to consider more decision factors and features being adapted to this model.

REFERENCES

[1] Z. Sheng, C. Mahapatra, C, Zhu, V.C.M. Leung, "Recent Advances in wireless sensor networks towards efficient management IoT" IEEE Access 2015, 3,pp. 622–637.

[2] L.M. Borges, F.J. Velez, A.S. Lebres, "Survey on the Characterization and Classification of Wireless Sensor Network Applications" IEEE Commun. Surv. Tutor. 2014, 16, pp. 1860–1890.

[3] T. Loh, K. Liu, F. Qin, H. Liu, "Assessment of the adaptive routing performance of a Wireless Sensor Network using smart antennas" IET Wirel. Sens. Syst. 2014, 4, pp. 196–205.

[4] Y. Wang, M. Motani, H.K. Garg, Q. Chen, T. Luo, "Cooperative Multichannel Directional Medium Access Control for Ad Hoc Networks" IEEE Syst. J. 2017, 11, pp. 2675–2686.

[5] A. Chau, J Dawson, P. Mitchell, T.H. Loh, "Virtual Sensing Directional Hub MAC (VSDH-MAC) Protocol with Power Control" *Electronics* 2020, *9*, pp. 1219.

[6] H. Zhuochuan, S. Chien-Chung, C. Srisathapornphat, C. Jaikaeo, "A busy-tone based directional MAC protocol for ad hoc networks", In Proceedings of the MILCOM 2002, 1232, pp. 1233–1238.

[7] D. Duc Ngoc Minh, L. Huong Tra, K. Hyo Sung, H. Choong Seon, C. Jongwon, "Multi-channel MAC protocol with Directional Antennas in wireless ad hoc networks", In Proceedings of the 2015 International Conference on Information Networking (ICOIN), pp. 81–86.

[8] H. Xu, L. Lei, "A MAC protocol based on busy tone for ad-hoc networks with beamforming antennas", In Proceedings of the 3rd International Conference on Systems and Informatics (ICSAI), 2016, pp. 659–6630

[9] B. Sugiarto, R. Sustika, "Data classification for air quality on wireless sensor network monitoring system using decision tree (DT) algorithm" In Proceedings of the IEEE 2nd International Conference on Science and Technology-Computer (ICST), 2016, pp. 172-176.

[10] C. Xu, Z. Xiong, G. Zhao, S. Yu, "An energy-efficient region source routing protocol for lifetime maximization in WSN" IEEE Access 2019, 7, pp. 135277–135289.

[11] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001).

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, 2011, 12, pp. 2825–2830.

[13] X. Zheng, P. Li, Z. Chu and X. Hu, "A Survey on Multi-Label Data Stream Classification," in IEEE Access,2020, vol. 8, pp. 1249-1275.

[14] RiverBed. Available online: https://www.riverbed.com/ (accessed on 27 July 2020).

[15] "Crossbow Technology Inc MICAz datasheet Document, Part Number: 6020-0060-04 Rev A.,"