IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

Efficient DAG Task Scheduling Algorithm for Wireless Sensor Networks

Mandeep Kaur[1*], Balwinder Singh Sohi[2]

[1]Research Scholar, IKG Punjab Technical University, Jalandhar, Punjab, India
[2]Director, Chandigarh Group of Colleges, Mohali, Punjab, India

Corresponding Author: Mandeepkaur03@gmail.com

Available online at: www.ijcseonline.org

Accepted: 20/Dec/2018, Published: 31/Dec/2018

Abstract- WSN (Wireless Sensor Network) has to be systemized to being resource effective and extensible. The concept of task scheduling requires less task completion time, less energy consumption and proper utilization of energy with fewer energy nodes. DAG (Directed Acyclic Graph) is considered for WSN organization as of the routing redundancy for the root. This research paper has aimed to enhance the performance of DAG in WSN. For the improvement, Cuckoo Search as an optimization algorithm and Support Vector Machine as a classification algorithm is utilized. For the evaluation of the proposed work, QoS parameters, such as Throughput, PDR (Packet Delivery Ratio), Energy Consumption, Network Lifetime and Delay are considered. The proposed algorithm stood very positive results for each QoS parameter. The results are compared on the base of with and without the proposed architecture.

***Keywords*: WSN, DAG, Cuckoo Search, Support Vector Machine**

I. INTRODUCTION

A wireless sensor network (WSN) can be defined as a wireless network consisting of devices being spatially distributed by utilizing sensors for monitoring physical and environmental conditions.WSN is constructed with nodes that are utilized for Computation_Energy serving the surrounds, namely, humidity, temperature, pressure, vibration, sound, and position etc. These nodes could be utilized in many functions (real-time) for performing a variety of tasks like a neighbor node innovation, smart detecting, manipulating and storing data, target tracking, data collection, synchronization, node localization, monitor and control, and efficient routing within nodes along with a base station. The wireless sensor network can be categorized by utilizing:

1. Transmission media based networks include wired sites where communication occurs with the assistance of wires and Wi-Fi sites where communication occurs wirelessly.
2. System Sizing based sites include MAN, personal computer community and WAN [1].

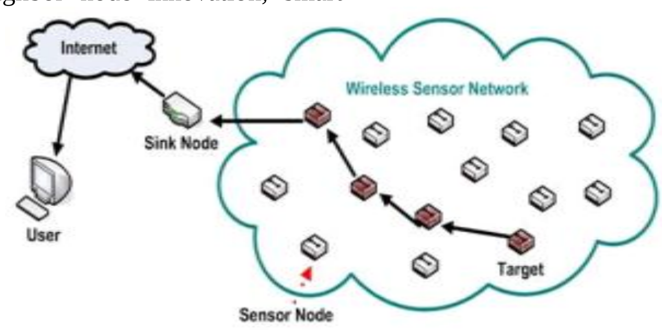

Figure Error! No text of specified style in document.**: Wireless Sensor Network**

Figure 1 shows the classifications of sensor networks (wireless) and is defined below:

i. Sensing unit: Sensing unit comprises of a sensor and ADC (Analog to digital converters) that transfers analogy signal obtained by the sensor to a digital signal. It helps in the conversion of physical anomalies to an electrical signal.

ii. Processing unit: Processing unit comprises of microcontroller or microprocessor that control sensors, communication protocols execution with the processing of algorithms (signal) on the sensor data (composed).

iii. Transmission Unit: It takes the information from CPU and later transfers that to user end in the world outside.

iv. Power Unit: In WSN, battery power is considered as the main source of energy. The battery power is supplied to the sensor node by power units [2].

1.1.1 Categorization of sensor network (wireless)

Wireless systems could possibly be categorized into two categories:

i. Infra type structure Network

Within commercial infrastructure primarily based on network, communication takes place just between the Wi-Fi nodes and also the entry points. The particular communication will not occur between the Wi-Fi nodes.

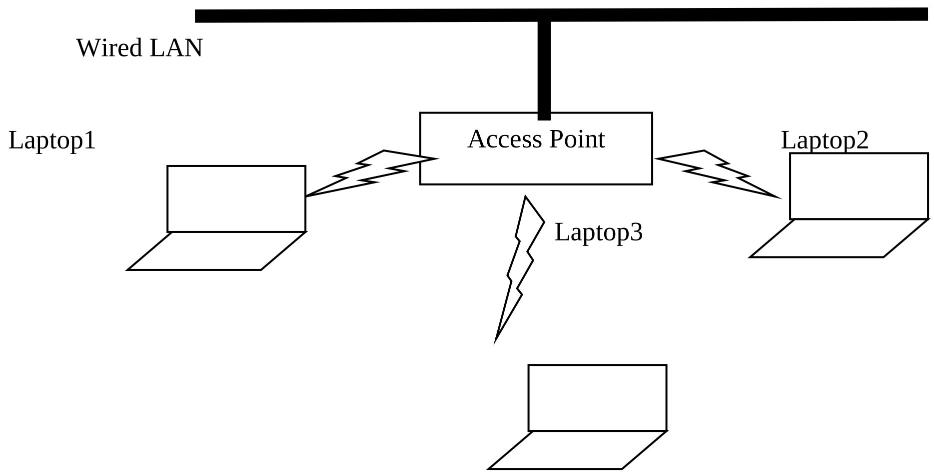

Figure 2: Infrastructure Network

The access point is used to manage the particular medium entry moreover gives it functions while establishing a link between the Wi-Fi and wired sites. In this kind of network, fixed base locations are utilized. The example of infrastructure based network is cell phone networks. It can be a centralized method which is certainly governed through the controller just like a router. The most negative aspect in this method can be that if the controller isn't able, the whole method may crash.

ii. Infrastructure-less Networks

The particular infrastructures less network do not need any kind of commercial infrastructure. In this community, every node will communicate by using distinct nodes. Therefore, in this network, not an entry should be applied pertaining to control medium entry. In this community, all the nodes had to behave as routers in addition to every node being efficient at movements

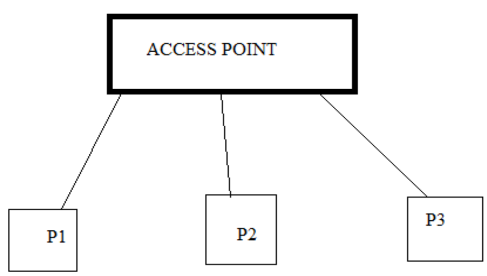

Figure 1 Infrastructure-less Network

The following Number exhibits an easy expert to check community using 3 nodes. To help ahead packets between the outside nodes, the middle node may be used being a router.

Example of the infrastructure-less network is Ad-hoc Network [5].

A Wireless Sensor System (WSN) consists a node group coupled with little functionality. An invisible detector community comprises of one or numerous drain nodes and it helps to gather nodes along with central procedure exchange and process of space storage. A detector node is typically power-driven with battery pack and usually ends up splitting into three functioned products (primary): realizing unit, communication unit, and a processor unit. The recent advancement in micro-electro-mechanical machines, Wi-Fi machines along with digital physics has proved as a boon for detector nodes. As a result, blossoming prospect associated with WSNs can be utilized for the practical purpose.

i. Energy saves: Agents of Mobile discard fusion of data and sensor node processing to the agent; For dislocating agents there are already made data. This may reduce network data embattled visitors. It also helps to save the bandwidth of the network, reduce hold off (end to end) and enhance the response of support. Therefore, the mobile agent may lessen energy consumption and enhance the lifetime of the network.

ii. Easier network procedure: WSN is an application-oriented network, which strains the need associated with other needs to be accepted by network protocols only. (From application layer to network layer data link layer). Systems protocols are available all through the reduce tiers joined with the a node software program.

iii. Complicated protocols are cumbersome for advancement and it makes network breakdowns easy. The agent of mobile may know users need and may summarize the network procedures. Needs of users are essentially customized (the real agent is customized and network protocols remain the same).

iv. Flexibility and independence: the Mobile agent may be changed (a new agent may be inserted in the network at any time). Also, a single sensor node could possibly work with numerous agents, at the same time. As a result, we can say, network mobility and versatility are considered beneficial. Furthermore, mobile agents can figure out modifications of the natural environment, reply rapidly, as well as maintain the perfect system conditions [6-8].

1.1.2 Applications of Wireless Sensor Network

WSN consists of a variety of sensors, namely, seismic, infrared, visual, radar, thermal, magnetic etc. that monitors the broad range of ambient situations. Basically, sensor nodes are utilized for event ID, local control of actuators, constant sensing and event detection. The WSN includes military, home, environment and another commercial area

I. Scheduling Architecture in WSN

a) Static vs. Dynamic Scheduling

In Static scheduling, all the information about Computation_Energy and available resources is known in advance, after which the job is assigned to resources and in dynamic scheduling, jComputation_Energy are allocated to resources at runtime by the scheduler [23]. It is more flexible than static scheduling, helps in determining the runtime in advance. The overhead is more as compared to static scheduling.

b) Independent vs. Workflow Scheduling

In Independent Scheduling, the execution of tasks is done independently whereas, in Workflow Scheduling, the tasks are inter-dependent [24]. Dependency means there are priority orders for the tasks, means a task cannot initialize till every of its parent are finished. Workflows are illustrated as DAG (Directed Acyclic Graph) notation. Every task may start its execution simply when every preceding task in DAG is already done.

II. DAG Task Model

The directed graph having no cycles is known as Directed Acyclic Graph (DAG). A workflow application is shown by DAG, G=(V, E), in which V is v nodes set and every node vi€ V denotes an application task, with instructions that must be executed on some machine (processors). E is the e-communication edges set among tasks node,every(i, j)€E denotes task dependency constraint i.e., the task n_i must finish its task before the start of task for its execution.

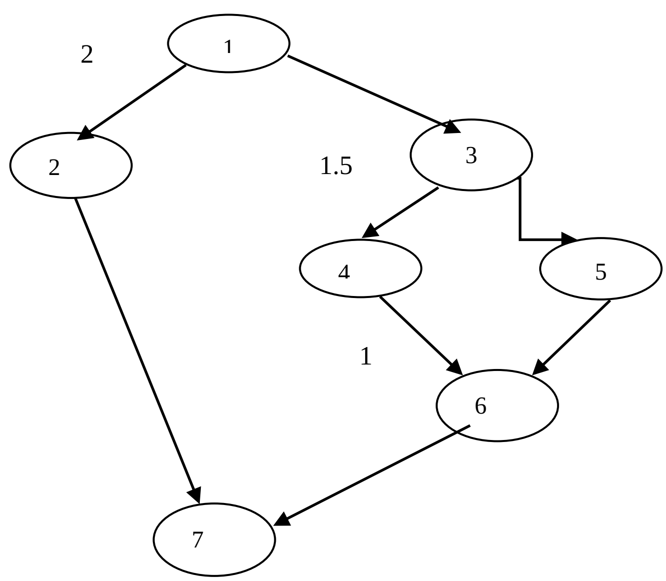

Figure 4: Directed Acyclic Graph Example

Figure 4 is describing the nodes denoting the tasks and the edges are defining the task dependencies whereas the edge weight is the waiting time for the successor's tasks to execute only when the predecessor tasks are finished.

a) List Scheduling Heuristics

The objective of list scheduling is to develop an ordered list of tasks by allocating them several priorities and next, constantly perform the subsequent two steps till a suitable schedule is obtained [31].

- Choose from the list, the task with the greatest priority for scheduling.
- Choose a resource to accommodate the task.
- If no resource could be found, the next task in the list is selected.

The priorities are executed statically before the beginning of the scheduling process. The initial step decides the task with more priority; the next step chooses the best probable resource. A few known strategies for list- scheduling are:

- HLF (Highest level first algorithm)
- LP (Longest path algorithm)
- Critical path method
- Heterogeneous Earliest Finish Time (HEFT)

III. PROPOSED WORK MODEL

The proposed work model aims to improve the DAG performance in the Wireless Network by applying Cuckoo Search and Support Vector Machine. The Cuckoo Search sets the precedence of packets of data to be transferred. The algorithm is as follows

Algorithm 1: Application of CS

1. Input : Packet_Sequence(PS) Output: Optimal Packet Stream
2. Total_Egg_Slots=4;
3. Divide the BS into Egg_slot Time Segments. // Dividing the packet stream into frames
4. eggsperslot=(duration)/Total_Egg_Slots;
5. initialcounter=1; // Taking a counter to keep optimal
6. // eggs only
7. finalcounter=eggsperslot
8. foreach egg in Total_Egg_Slots
9. Eggcounter=initialcounter:finalcounter // The
10. //eggcounter will start from 1 and will go to 4
11. Eggsinslot=BS(Eggcounter);// Extracting data from Bit
12. //Stream
13. TotalEggs=BS;
14. Fitvalue=CuckooFitness(Eggsinslot);// Passing the data to the fitness function of Cuckoo Search
15. End For

The proposed Cuckoo Search algorithm sets the priority schemes for the packets of data. The packets of data carry a value tag along with them. The proposed algorithm establishes a fitness function which covers the average bit tag value and multiplies it with cuckoo behavior and checks it with the bit tag value. If the bit value passes the Cuckoo Search Algorithm test then it is cross-validated by SVM and then it follows the DAG structure for processing. The structure of SVM is as follows.

Support Vector Machine, (SVM) is an algorithm that was developed for pattern classification but has recently been adapted for other uses, such as finding regression and distribution estimation. It has been used in many fields such as bioinformatics and is currently a very active research area in many universities and research institutes [56]. Even if SVM can be useful for optimizing numerous problems like regression, classification of data remains the main problem. In machine learning, the chore of figuring out a category from supervised training data is known as Sup1rvised Learning. In supervised learning, the training data comprises of an array of training examples, where every example consists of an input along with output. A guided Algorithm examines the data of training and then predicts the correct output categorization for given data-set input. Support Vector Machine (SVM) was introduced by Vapnik in 1979, and it is primarily defined as a supervised learning approach that utilizes a subcategory of training point which is also acknowledged as support vectors in order to categorize diverse entities. SVM basically searches the best possible linear decision surface concerning the binary classes. The biased arrangement of the support vectors is known as decision surface. We can also say that the nature of the margin among the two classes is decided by the support vectors.

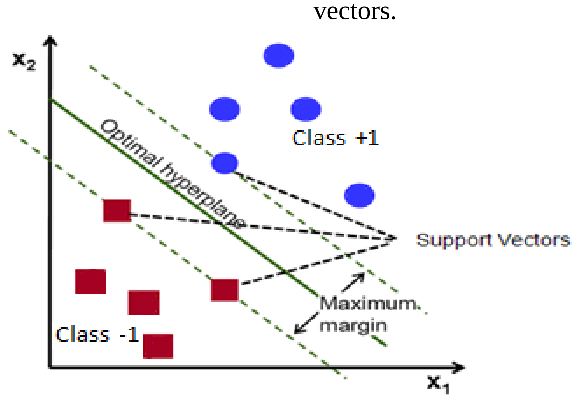

Figure 5: Support Vector Machines

Figure 5 only shows the case of 2-dimension where points of data are linearly separable. The red boxes and blue circle are the support vectors that are primarily the Computation_Energywhichsupports hyperplane on both sides. The hyperplane is simply defined as a line in 2D, whereas in 3D it defined as plane [57]. In greater dimensions (i.e. larger than 3D), it is entitled as a hyperplane. SVM aid to discover a hyperplane or unscrambling boundary that can isolate two classes. The margin is well-defined as the space among the hyperplane and the adjoining data point. If that space is doubled up, it would be identical to the margin.

Internally, SVMs are quadratic optimizers that search for a hyperplane that best separates the classes. A maximum margin SVMs strives to increase the distance of this hyperplane to the Support Vectors. i.e. examples from different classes that fall on both sides of the hyperplane.

Support Vector Machines are versatile, for different decision function. SVMs are very effective when very high dimensional spaces are concerned. Also, when the number of dimensions turns out to be bigger than the current number of samples, then the SVM's appeared to be very effective.

Algorithm 2: The Cross Validator()

Inputs: COMPUTATION_ENERGY, Allocated_Processors Output: Cross Validated Data (CVD)

1. CVD= [ ] // Initializing empty array
2. Range=[0: Allocated_Processors.length];
3. For KI=0: Range // Taking each time frame
4. Data_to_Validate=COMPUTATION_ENERGY[KI , :] // All the values of the time frame
5. Target_Set=Ones[Data_to_Validate.Length];
6. The foreach bit in Data_to_Validate
7. Processor_position=bit.positionvalue;
8. Target_Set[Processor_position]=2; // All the processor values are initialized with label set 1 and the current bit value //is labelled as 2
9. SVMStruct=Train_Svm(Data_to_Validate,Target _Set, 'Kernel Type=polynomial') // Training SVM
10. //with Training set and its corresponding label with the polynomial type of kernel
11. TestData=Data_to_Validate ; // Keeping the test data same as that of training data for supervised //learning
12. OutputLabel= SVMClassify(SVMStruct, TestData) ;
13. If OutputLabel[Processor_position]== Target_Set[Processor_position] // If the output label of the test data is same as //that of the training label then keep the data as it is
14. CVD [KI, Processor_position]= bit;
15. Else
16. Apply Algorithm 1 for a bit of this time stream
17. End If
18. End For
19. End For
20. Return CVD
21. End Algorithm

The processor values are cross-validated utilizing the energy consumption of the allocation and the total load on each processor. It is a supervised learning method and hence the training structure would be serving the test data as well. The test data will be equal to the training data only. Hence if the target value of the simulation is not equal to the training value, then the cross-validation fails and hence re allocation of the packet of data will be placed. The cross-validation structure reduces the load from computation and improves the overall network presentation.

IV. RESULTS AND EVALUATION

The evaluations of the results are made on the base of the following parameters.

a) Throughput : $\frac{Total_{received}}{Time_{transfer}}$
b) Packet Delivery Ratio(PDR) : $\frac{Total_{receieved}}{Total_{transferred}}$
c) Energy Consumption: $Total_{Consumed_{Energy\ per\ time\ frame\ in\ mj}}$
d) Delay: $Total_{produced_{latency}} per\ iteration$

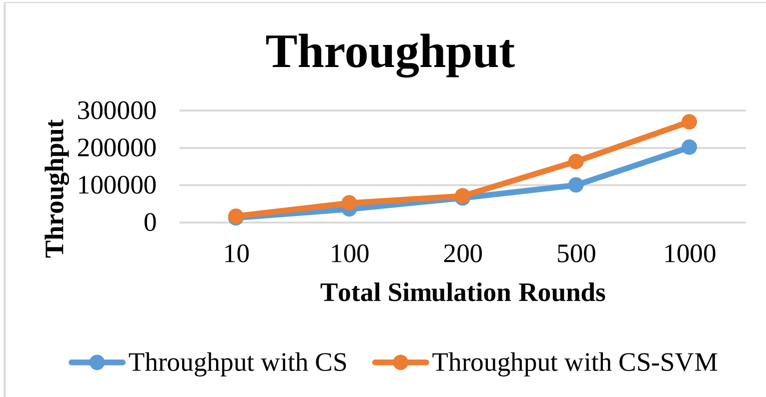

Figure 6: Throughput

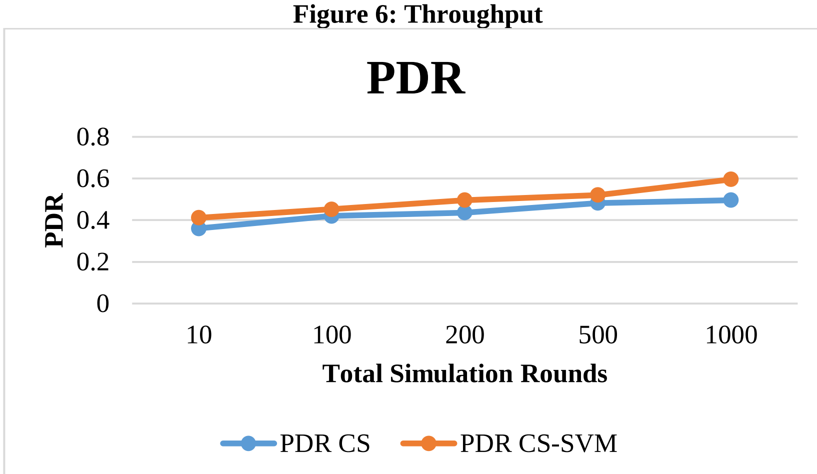

Figure 7: PDR

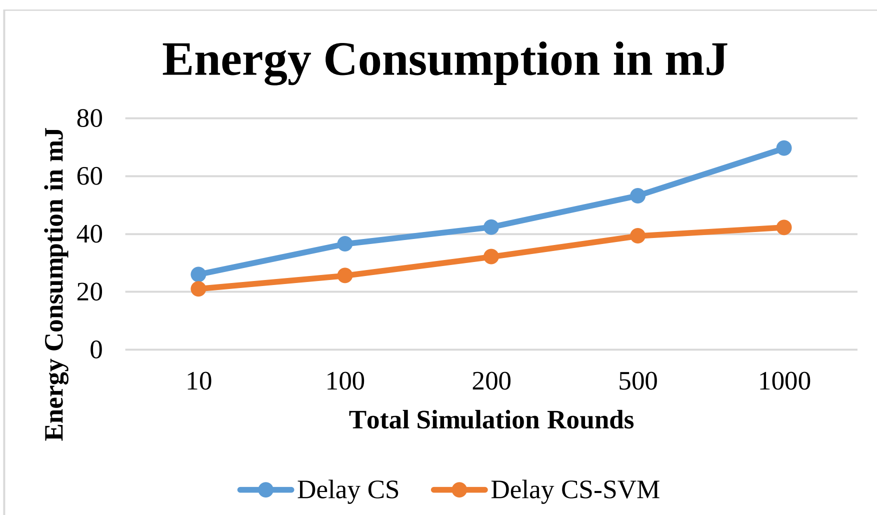

Figure 8: Energy Consumption in Mj

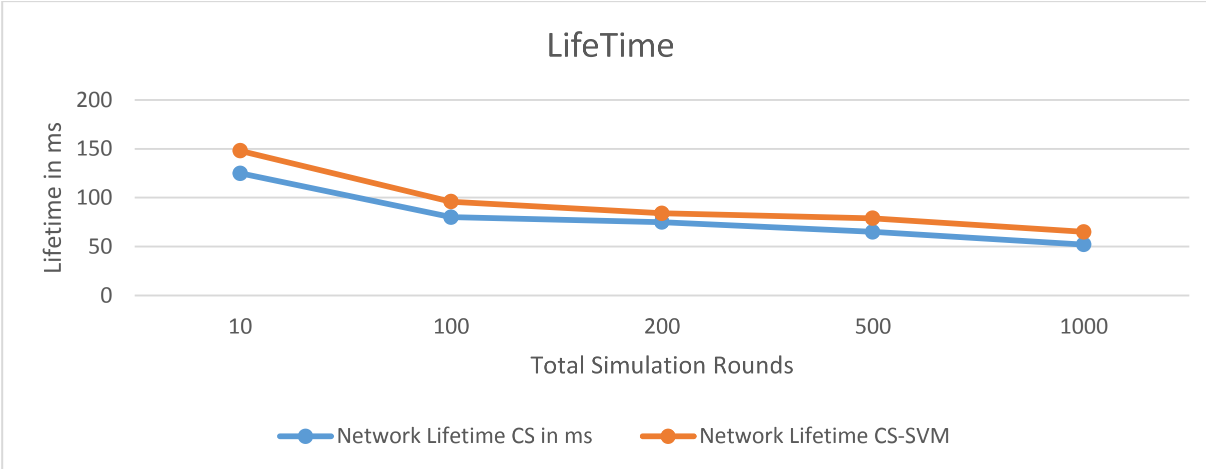

Figure 9: Network Lifetime in ms

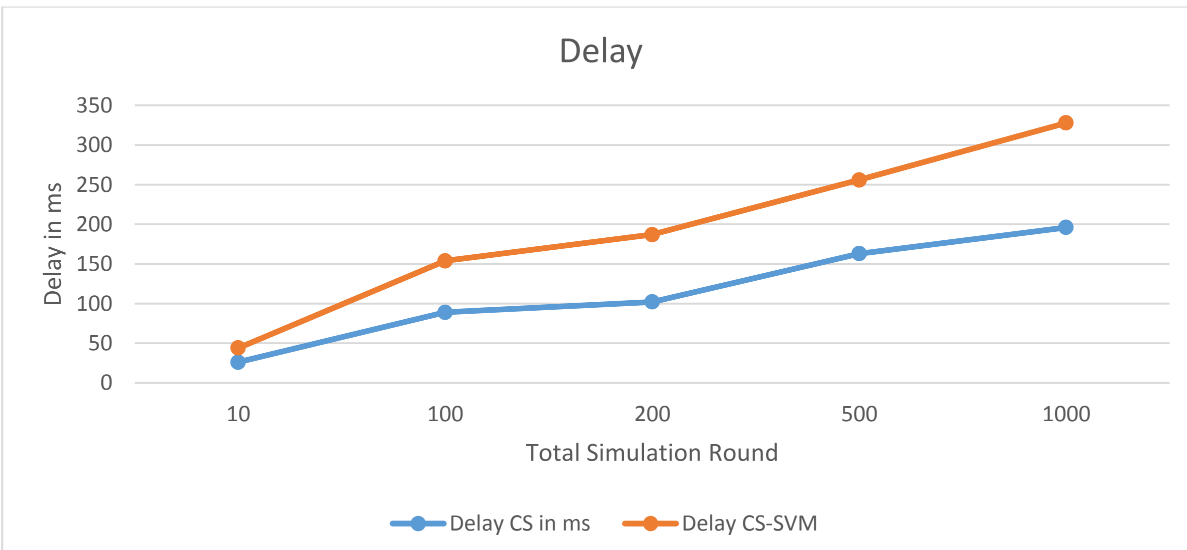

Figure 10: Delay in ms

The maximum throughput attained by the proposed algorithm is 269663 whereas if the SVM is missing from the proposed architecture then it results into 201561. The percentage growth is $\frac{269663-201561}{201561}$= 33 %. In a similar fashion, PDR and Energy Consumption both stand a great improvement. The maximum PDR is .5964 for the proposed structure where if SVM is missing from the proposed structure then it comes to .4966. The proposed structure delay is 42.25 milliseconds whereas the delay without SVM is 69.65. In a similar fashion, Delay and Lifetime are also calculated. The maximum delay produced while utilizing CS is 196 whereas compared to CS it is 132 ms. The Lifetime for the same architecture is 148 at max whereas 125 ms for CS.

V. CONCLUSION

DAG is a cyclic graph structure which is when deployed in WSN does wonder for the delivery of the packets of the network. The proposed algorithm utilizes Cuckoo Search (CS) and Support Vector Machine (SVM) as optimization and cross-validation algorithm. The cross-validation structure ensures the correct allocation of the packets of data to the processing units. The proposed structure demonstrates an improvement of around 30-33 % for the evaluated QoS parameter structure Throughput, Energy Consumption and Delay in the packet delivery.

REFERENCES

[1]. Elson and A. Parker. Tinker, “A tool for designing data-centric sensor networks”, *Information Processing in Sensor Networks, The Fifth International Conference*, pp. 350-357, 2006.

[2]. A. Razaque, M. Abdulgader, C. Joshi, F. Amsaad and M. Chauhan, “P-LEACH: Energy efficient routing protocol for Wireless Sensor Networks,” *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2016, pp. 1-5.

[3]. R. Ramadan, K. Abdelghany, and H. El-Rewini, “SensDep: a design tool for the deployment of heterogeneous sensing devices”, In Second IEEE Work- shop on Dependability and Security in Sensor Networks and Systems, 2006.

[4]. KP Ferentinos and TA Tsiligiridis, “Heuristic dynamic clustering in wire- less sensor networks towards uniform sensing”, 15th IST Mobile & Wireless Communications Summit, Myconos, Greece, June, 2006.

[5]. Eduardo F. Nakamura, Antonio A. F. Loureiro, and Alejandro C. Frery, “Information fusion for wireless sensor networks: Methods, models, and classifications”, ACM Computing Survey, vol.39, 2007.

[6]. Vieira, Marcos Augusto M., Claudionor N. Coelho, D. C. Da Silva, and José Monteiro da Mata. "Survey on wireless sensor network devices." In Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference, vol. 1, pp. 537-544. IEEE, 2003

[7]. Alasmary, W.; Weihua Zhuang, “The Mobility Impact in IEEE 802.11p Infra structureless Vehicular Networks,” in Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd , vol., no., pp.1-5, 6-9 Sept. 2010

[8]. Sandra Sendra, Jaime Lloret, Miguel García and José F. Toledo, “Power saving and energy optimization techniques for Wireless Sensor Networks”, IEEE, Vol.6 pp. 439-452, 2011.

[9]. Akyildiz, I.F.; Weilian Su; Sankarasubramaniam, Y.; Cayirci, E., “A survey on sensor networks,” in Communications Magazine, IEEE, vol.40, no.8, pp.102-114, Aug 2002.

[10]. S. Xu and T. Saadawi, “Does the IEEE802.11 MAC protocol work well in multi hop wireless ad hoc networks?", IEEE Commun. Mag., vol. 39, pp.130 -137 2001.

[11]. Toumpis, Stavros; Goldsmith, A.J., “Capacity regions for wireless ad hoc networks,” in Wireless Communications, IEEE Transactions on , vol.2, no.4, pp.736-748, July 2003.

[12]. M. Kaur and A. Sharma, “Routing in WSN network using neural Network (NN) and SPEED protocol,” *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Greater Noida, India, 2016, pp. 161-167.

[13]. X. Wang, H. Cheng and Y. Yao, “Addressing-Based Routing Optimization for 6LoWPAN WSN in Vehicular Scenario,” in *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3939-3947, May15, 2016.

[14]. U. Raza, A. Camerra, A. L. Murphy, T. Palpanas and G. P. Picco, “What does model-driven data acquisition really achieve in wireless sensor networks?,”*2012 IEEE International Conference on Pervasive Computing and Communications*, Lugano, 2012, pp. 85-94.

[15]. S. Ghosh, S. Mondal and U. Biswas, “Fuzzy C Means based Hierarchical Routing Protocol in WSN with Ant Colony Optimization,” *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Bangalore, India, 2016, pp. 348-354.

[16]. A. Sharma and P. Lakkadwala, “Performance comparison of reactive and proactive routing protocols in Wireless Sensor Network,” *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, Noida, 2014, pp. 1-6.

[17]. B. Paul, K. A. Bhuiyan, K. Fatema and P. P. Das, “Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network,” *2014 International Conference on Computational Intelligence and Communication Networks*, Bhopal, 2014, pp. 364-369.

[18]. A. Komathi and M. Pushparani, “Trust performance of AODV, DSR and DSDV in wireless sensor networks,” *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, Coimbatore, 2014, pp. 423-425.

[19]. S. Li, W. Dai and S. Zhang, “NS2 Enabled Multi-Routing-Protocol Simulating of Wireless Mesh Networks,” *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, 2012, pp. 1-4.

[20]. H. Jin, L. Ren and Y. Ding, “A Hierarchical Wireless Routing Protocol for Multi-Channel Multi-Interface Ad Hoc Networks,” *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, 2011, pp. 1-4.

[21]. L. He, J. Huang and F. Yang, “A noval hybrid wireless routing protocol for WMNs,” *2010 International Conference on Electronics and Information Engineering*, Kyoto, 2010, pp. V1-281-V1-285.

[22]. A. Rahimi and A. Harati, “CGSR features: Toward RGB-D image matching using color gradient description of geometrically stable regions,” *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Rasht, 2015, pp. 1-6.

[23]. H. Raza, P. Nandal and S. Makker, “Selection of cluster-head using PSO in CGSR protocol,” *2010 International Conference on Methods and Models in Computer Science (ICM2CS-2010)*, New Delhi, 2010, pp. 91-94.

[24]. T. P. Venkatesan, P. Rajakumar and A. Pitchaikkannu, “Overview of Proactive Routing Protocols in MANET,” *2014 Fourth International Conference on Communication Systems and Network Technologies*, Bhopal, 2014, pp. 173-177.

[25]. L. Mejaele and E. Oketch Ochola, “Effect of varying node mobility in the analysis of black hole attack on MANET reactive routing protocols,” *2016 Information Security for South Africa (ISSA)*, Johannesburg, 2016, pp. 62-68.

[26]. L. Mejaele and E. Oketch Ochola, “Effect of varying node mobility in the analysis of black hole attack on MANET reactive routing protocols,” *2016 Information Security for South Africa (ISSA)*, Johannesburg, 2016, pp. 62-68.

[27]. N. Bilandi and K. Kaur, “Performance analysis of propagation structures on DSR routing protocol in mobile ad-hoc networks,” *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, 2016, pp. 445-449.

[28]. B. K. Joshi and M. Soni, “Security assessment of AODV protocol under Wormhole and DOS attacks,” *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Greater Noida, India, 2016, pp. 173-177.

[29]. A. Sharma and R. Kumar, “Performance comparison and detailed study of AODV, DSDV, DSR, TORA and OLSR routing protocols in ad hoc networks,” *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Waknaghat, India, 2016, pp. 732-736.

[30]. K. Madhuri, N. K. Viswanath and P. U. Gayatri, “Performance evaluation of AODV under Black hole attack in MANET using NS2,” *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, 2016, pp. 1-3.

[31]. Y. Sun, F. Jiang, T. Liang and C. Phillips, “Enhanced Associativity-Based Multicast Routing protocol in CR Mobile Ad hoc Networks,” *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, Washington DC, 2014, pp. 1233-1237.

[32]. B. Preveze and A. Şafak, “Associativity Tick Averaged Associativity based Routing (ATAABR) for real time mobile networks,” *2009 International Conference on Electrical and*

Electronics Engineering - ELECO 2009, Bursa, 2009, pp. II-208-II-212.
[33]. R. Dube, C. D. Rais, Kuang-Yeh Wang and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," in *IEEE Personal Communications*, vol. 4, no. 1, pp. 36-45, Feb 1997.
[34]. A. Chandra and S. Thakur, "Performance evaluation of hybrid routing protocols against network layer attacks in MANET," *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, 2015, pp. 239-244.
[35]. C. Y. Aung, P. H. J. Chong and R. J. Cai, "Hybrid opportunistic routing in highly dynamic MANET," *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, Shanghai, 2014, pp. 1-6.
[36]. S. A. Alharthi and P. A. Johnson, "Threshold sensitive heterogeneous LEACH protocol for wireless sensor networks,"*2016 24th Telecommunications Forum (TELFOR)*, Belgrade, 2016, pp. 1-4.