Int. J. Advanced Networking and Applications
Volume: 02, Issue: 03, Pages: 693-698 (2010)

An Efficient Congestion Avoidance Scheme for Mobile Healthcare Wireless Sensor Networks

R.Gunasundari
Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry.
Email: r_gunasundari@rediffmail.com
R.Arthi
Department of Electronics and Instrumentation Engineering, St. Joseph Engineering College, Chennai.
Email: rg_barathy@yahoo.co.in
S.Priya
Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry.
Email: r_gunasundari@rediffmail.com

ABSTRACT

Wireless Sensor Networks (WSN) in healthcare environment continuously monitors critically ailing patients. Congestion is one of the major challenges in WSN; it causes overall channel quality to degrade, loss rates to raise, leads to buffer drops and increased delays, and tends to be grossly unfair toward nodes whose data has to traverse a larger number of radio hops. Congestion avoidance deserves first place in healthcare environment. The problem of congestion in the nodes of healthcare WSN is addressed using a Learning Automata (LA).The Learning Automata Based Congestion Avoidance Scheme (LACAS) can counter the congestion problem efficiently. LACAS intelligently learns from the past and improves its performance significantly as time progresses awnd it is suitable only for stationary environments. Mobile healthcare provides accessible services that are welcoming to homeless people who cannot go to fixed-site clinics, so that mobility for nodes in healthcare WSN is needed. Congestion avoidance in mobile healthcare WSN is addressed by implementing LACAS in the nodes.

Keywords - Congestion avoidance, Mobile healthcare applications, performance comparison

Date of Submission: September 3, 2010 Date of Acceptance: October 29, 2010

1. INTRODUCTION

Recent advancement in wireless communications and electronics has enabled the development of low-cost sensor networks. The sensor networks can be used for various application areas (e.g., health, military, home). WSN typically consists of small devices called sensor nodes that are capable of sensing, gathering, storing and transmitting information. One of the popular application domain of WSNs is healthcare, specifically, the remote monitoring of the conditions of ailing patients [1]. Congestion is one of the major challenges in WSNs. One of the popular congestion control mechanism is energy efficient congestion control scheme for sensor networks called CODA (congestion detection and avoidance). Congestion is detected based on the queue length of packets at the intermediate nodes. CODA comprises of three key mechanisms – congestion detection, open loop hop-by-hop backpressure and closed loop multi-source regulation. In open loop hop-by-hop backpressure, the source gets the backpressure signals depending on the local congestion state. In closed loop multisource regulation, the source gets an ACK from the sink and when the congestion occurs, the sink stops sending ACKs to the source [2]. Another popular congestion control mechanisms for WSNs. is Adaptive Rate Control (ARC) [3]. In ARC, the Constant Bit Rate (CBR) of the source and the intermediate nodes are changed, whenever a node receives a feedback regarding congestion from its child node. It follows the Additive Increase and Multiplicative Decrease (AIMD) algorithm, which is, essentially, a rate-based mechanism, in which the intermediate node increases its sending rate by a constant, α, when its parent node forwards the packet successfully. Otherwise, the intermediate node multiplies its sending rate by a factor, β, where $0 < \beta < 1$. LACAS can counter the congestion problem in healthcare WSNs effectively [4]. The primary objective in using this approach is to adaptively make the processing rate (data packet arrival rate) in the nodes equal to the transmitting rate (packet service rate), so that the occurrence of congestion in the nodes is seamlessly avoided. LACAS has been implemented for stationary environment. The proposed work is to address LACAS for mobile environment.

2. RELATED WORK

The congestion problem is avoided by placing some simple autonomous learning machines, called "automata"(can be construed as small pieces of code capable of taking "intelligent" actions), at each of the nodes of the network that are capable of controlling the rate of flow of data at the intermediate nodes based on probabilistically how many packets are likely to get dropped if a particular flow rate is maintained. An automaton stationed at each node "learns" from the past behavior and chooses a "better" data flow rate that is likely to avoid congestion from occurring in the network [3], [4], [5].

2.1 Learning Automaton

The theory of LA centers on the notion of an "*automaton,*" which is a self-operating machine or a mechanism that responds to a sequence of instructions in a certain way, so as to achieve a certain goal. The automaton either responds to a pre-determined set of rules, or adapts to the environmental dynamics in which it operates. The term "*learning*" refers to the act of acquiring knowledge and modifying one's behavior based on the experience gained. The automata, then, attempt to learn the best action from a set of possible actions that are offered to them by the stationary environment in which they operate. The automata, thus, act as decision makers to arrive at the best action.

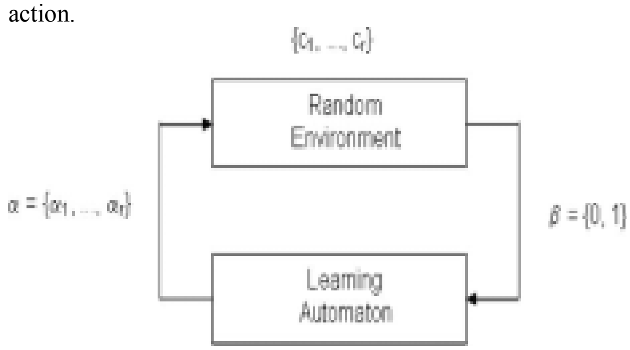


Figure.1 The Automaton-Environment feedback loop

2.1.1 Automaton

The Automaton, in our case, is, generically defined by a quintuple {A,B,Q,F(., .),G(.)}, where:

• $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of outputs or actions, and $\alpha(t)$ is the action chosen by the automaton at any instant t.

• B is the set of inputs to the automaton, $\{\beta_1, \beta_2, \ldots, \beta_r\}$. Here, $\beta(t)$ is the input at any instant t, while the set B can be finite or infinite.

• $Q = \{q_1(t), q_2(t), \ldots, q_s(t)\}$ is the set of finite states, where q(t) denotes the state of the automaton at any instant t.

• $F(., .) : Q\times B \rightarrow Q$ is a mapping in terms of the state and input at the instant t, such that, $q(t+1) = F[q(t), \beta(t)]$. It is called a transition function, i.e., a function that determines the state of the automaton at any subsequent time instant (t+1). This mapping can either be deterministic or stochastic, depending on the environment in which the automaton operates.

• G(., .): is a mapping function $G : Q \rightarrow A$, and is called the output function. Depending on the state at a particular instant, this function determines the output of the automaton at the same instant as: $\alpha(t) = G[q(t)]$. This mapping can, again, be considered to be either deterministic or stochastic, depending on the environment in which the automaton operates. Without loss of generality, G is deterministic.

2.1.2 The Environment

The Environment, E, typically, refers to the medium in which the automaton functions. The Environment possesses all the external factors that affect the actions of an automaton. Mathematically, an Environment can be abstracted by a triple {A, C, B}. A, B, and C are defined as follows.

- $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents a finite input set
- $B = \{\beta_1, \beta_2, \ldots, \beta_r\}$ is the output set of the environment, and
- $C = \{c_1, c_2, \ldots, c_r\}\}$ is a set of penalty probabilities, where element $c_{i,} \in C$ corresponds to an input action α_i.

The process of learning is based on a learning loop involving the two entities: the Random Environment (RE), and the LA, as described in Figure.1. In the process of learning, the LA continuously interacts with the Environment to process responses to its various actions. Finally, through sufficient interactions, the LA attempts to learn the optimal action offered by the RE. The actual process of learning is represented as a set of interactions between the RE and the LA.

The RE offers the automaton with a set of possible actions $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ to choose from. The automaton chooses one of those actions, say α_i , which serves as an input to the RE. Since the RE is aware of the underlying penalty probability distribution of the system, depending on the penalty probability $c_{i,}$ corresponding to α_i , it "prompts" the LA with a reward (typically denoted by the value '0'), or a penalty (typically denoted by the value '1'). The reward/penalty information (corresponding to the action) provided to the LA helps it to choose the subsequent action. By repeating the above process, through a series of Environment-Automaton interactions, the LA finally attempts to learn the optimal action from the Environment.

2.1.3 Action Probability Updating

Variable Structure Stochastic Automata (VSSA) are the ones in which the state transition probabilities are not fixed. In such automata, the state transitions or the action probabilities themselves are updated at every time instant using a suitable scheme. The transition probabilities and the output function in the corresponding Markov chain vary with time, and the action probabilities are updated on the basis of the input. VSSA depend on random number generators for their implementation. The action chosen is dependent on the action probability distribution vector, which is, in turn, updated based on the reward/penalty input that the automaton receives from the RE. The action probability updating scheme that we have designed is, essentially, a Linear Reward-Inaction Scheme (LRI) scheme. It is based on the principle that whenever the automaton receives a favourable response (i.e., reward) from the environment, the action probabilities are updated, whereas if the automaton receives an unfavourable response (i.e., penalty) from the environment, the action probabilities are unaltered.

3. An Automaton Stationed In A Node

An automaton, a simple autonomous machine (code) capable of making decisions, is stationed at every node in the network, as shown in Figure. 2.

What is noteworthy is that, only the nodes that act as intermediate nodes during the transmission of specific information will have their automata work for controlling congestion locally in that node. In other words, at any time instant, look at the network topology, the automata stationed in the intermediate nodes, and not the ones in the source nodes, will act as controllers of congestion of data arriving from source nodes. Moreover, note that each node is independent in the network in its approach for controlling the congestion. The algorithm is non-distributed in nature. In a distributed approach, the complexity of network will increase manifold and is also likely to decrease the reliability, since there are chances of the operations getting unsynchronized.

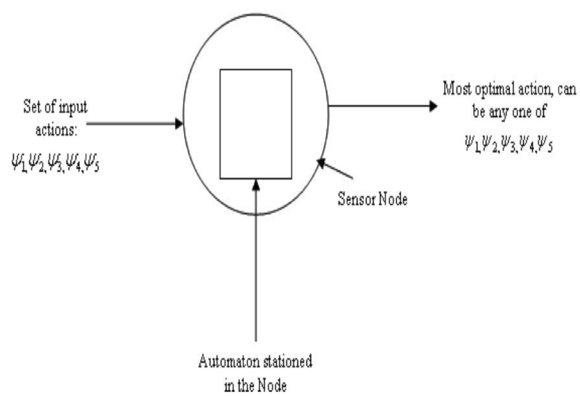


Figure.2.An automaton stationed in a node

For the input to the automaton at time, t = 0, the number of actions associated with an automaton to 5, which are based on the rate with which an intermediate sensor node receives the packets from the source node. Denote these actions by $\psi = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \}$ as shown in Figure.2. The rates, "ψ", that are taken as inputs to an automaton stationed in a particular node, are based on the number of packets dropped till then in the concerned node.

The most optimal action, at any time instant, among the set actions in a node, is decided by the number of packets dropped. To be precise, the rate of flow of data into a node for which there is the least number of packets dropped is considered to be the most optimal action. At any time instant, the choice of an action by the automaton, i.e., the rate at which data should flow into the corresponding node is rewarded / penalized by the environment. Initially, at t = 0, these actions have the equal probability (say, $P\psi_i$ (n)) of getting selected by the automaton. Let us assume that the automaton selects, ψ_1, initially, based on the probability values of all the actions at time t = 0. The chosen action, which maps to a certain rate of flow of data, which is predefined, then, interacts with the environment. The environment examines the action, $\psi 1$, and rewards/penalizes that action based on the packets dropped at the node. If the action, $\psi 1$, is rewarded, the probability value of is increased and the probability values of the other actions, i.e., ψ_2, ψ_3, ψ_4, ψ_5, are decreased as per the equations in (1) and (2)

$$P_{\psi_1}(n+1)\,P_{\psi_1}(n)+\frac{1}{\lambda}\left(1-P_{\psi_1}(n)\right) \quad (1)$$

$$P_{\psi_{2,3,4,5}}(n+1)=\left(1-\frac{1}{\lambda}\right)P_{\psi_{2,3,4,5}}(n) \quad (2)$$

If, in case, ψ_1 is penalized, the probability value corresponding to this action as well as for rest of the actions $\psi_{2,3,4,5}$, will remain unaffected. This is shown, mathematically, in Equations (3) and (4).

$$P_{\psi_1}(n+1)\,P_{\psi_1}(n) \quad (3)$$

$$P_{\psi_{2,3,4,5}}(n+1)=P_{\psi_{2,3,4,5}}(n) \quad (4)$$

The probability values associated with all the actions should be such that at every time instant the summation of their probabilities should equal unity. This is a fundamental law of probabilities. At the next time instant, again, the automaton will select an action based on the updated probability values. The selected action will interact with the environment again and will be rewarded or penalized accordingly. The probabilities of all the actions will be updated repeatedly in a continuous cycle.

This vicious cycle will continue until the most optimal action is selected, i.e., $\lim_{t\to\infty} P\psi_1 (n) = 1$, which means that the probability of most desirable action tends to unity as time tends to infinity. Let us assume that as $t\to\infty$, the automaton selects action, ψ_2, to be the most optimal action for the system. The sensor node will then emit the packets based on the rate corresponding to action ψ_2. Since this is the most optimal action selected by the automaton, the chance of the node getting congested reduces. For action, ψ_2, the number of packets dropped reduces to the minimum value out of the case if other actions were chosen. Also, it will transmit the data packets with the rate that corresponds to ψ_2^5. The above-mentioned approach is also followed at all the nodes of the network.

4. LACAS FOR MOBILE ENVIRONMENT

In the case of accidents and natural disaster, mobile healthcare is needed. Congestion avoidance in such mobile environment is very essential. In existing system, Learning automata congestion avoidance scheme is implemented only for stationary environment. In stationary healthcare environment all the sensor nodes are static, e.g. the source node (patient), destination node (doctor) and all intermediate nodes are static. But today in the case of accidents and natural disasters, the injured patients are carried to hospital through **ambulance**. Patients can remain under constant observation of expert physicians without being physically present at the hospital.Sensors that are fixed in the patient body monitors the patient condition, the collected information from the sensors are transmitted to the doctor, through the intermediate node, the doctor and the intermediate nodes can also be mobile, in such environment congestion avoidance is needed. LACAS is implemented in mobile WSN shown in below Figure.3

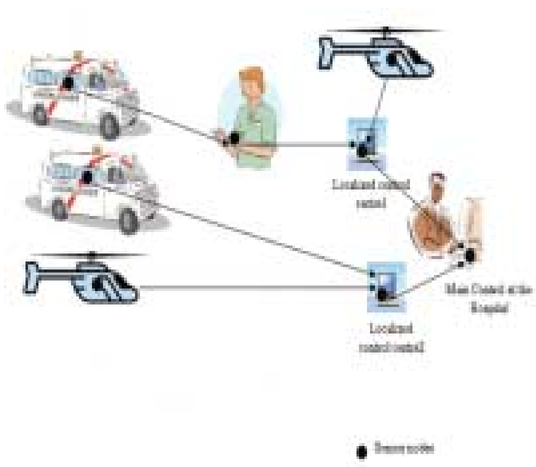


Figure.3 LACAS for mobile environment

4.1 LACAS ALGORITHM

In this proposed work LACAS algorithm is modified for mobile environment. Implementation of LACAS provides significant improvement in stationary environment. The problem of LACAS in mobile environment is overcome by implementing modified LACAS algorithm.

Principal Steps:

BEGIN

Step 1: Automaton will run only in the intermediate nodes. A loop is put to run the automaton in all the intermediate nodes.

Step 2: The automaton in the intermediate node, equates the packet arrival rate and the packet service rate to avoid queuing and delay. The intermediate node decides the rate at which the packets should be sent in order to get minimal loss, so that the probability of selecting an action from the set of actions is initialized.

Step 3: The intermediate node, selects an action randomly out of n actions present. Choose an action randomly out of a set of n actions as follows: =rand ( ) %n+1.

Step 4: When the selected action gets the favourable response from the environment, the probability of the response is updated in the node.

Step 5: When the number of packets received by the intermediate node, exceeds its threshold, fragmentation is done.

END

5. SIMULATION RESULTS

The performances of LACAS through simulations are evaluated using Global Mobile Simulator. The performances of LACAS for stationary and mobile environment is analysed and compared. The nodes are distributed to move randomly at a speed of 20m/s. Initially the source node is 1,3,10,15 and sink node is 2,6,7.

5.1 Setting and configuration

Simulation time	: 100 sec
Terrain dimension	:(40m, 40m)
No. of Nodes	: 16
Node placement	: Random
Propagation path loss:	Two ray
Temperature	: 290 K
Mac Protocol	: 802.11
Routing	: AODV

5.2. Performance metrics

5.2.1 Throughput Analysis

Throughput measures the average rate of a successful packet delivery over a communication channel Learning Automata (LA) congestion avoidance algorithm is implemented for stationary environment and mobile environment. The comparative result of LACAS with mobility and without mobility is shown in Figure.4 .With LACAS, the rate of flow of packets into and out of a node is controlled by an automaton. Throughput is increased by implementing LACAS. The packet received and sent ratio of LACAS for stationary environment is 0.8 and for mobile environment is 0.6 respectively. Comparative results show that throughput performance in mobile environment is less than that of the stationary environment due to the mobility of the nodes, moving at a speed of 20 m/s.

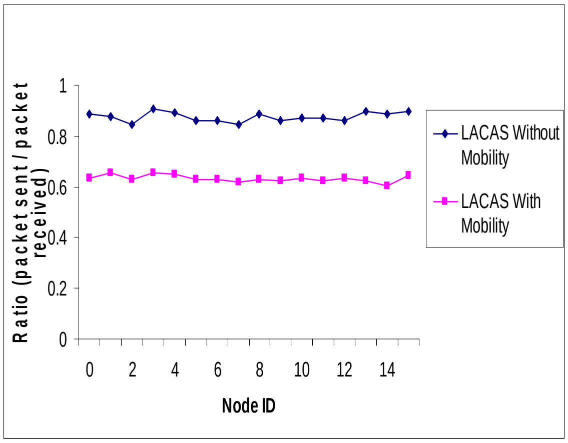


Figure 4 Throughput performance compared for stationary and mobile environment

5.2.2 Energy Statistics

Sensor nodes, especially, those used in healthcare applications, have tight constraints in terms of energy. Sensor nodes in pre-hospital and post-hospital environment closely monitoring the injured patients, it is difficult to replace the sources of energy in the nodes, when they get drained. In LACAS the rate of flow of packets is controlled in every node by automaton, due to that energy consumption is considerably reduced. Comparative results of energy statistics for stationary and mobile environment are compared in Figure.5. In stationary environment, transmission path is fixed and all the nodes are in on-state for all time, whether the packets is coming to it or not. In mobile environment, path is not predefined, whenever the packet is coming to a particular node then only it will turn to on-state, otherwise the node moves to sleep mode in order to save energy.

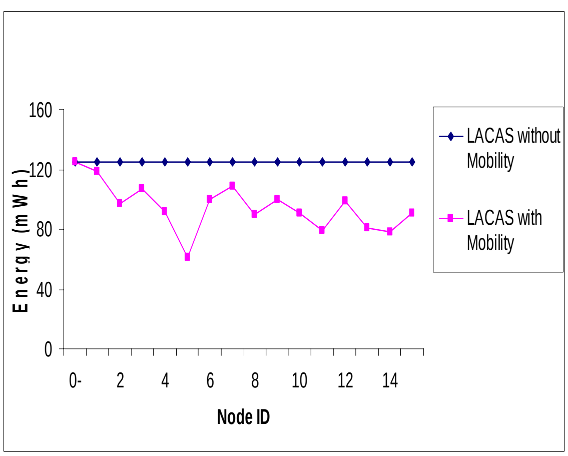


Figure 5 Energy statistics is compared for stationary and mobile environment

5.2.3 Collision Statistics

Collision statistics represents the number of collisions occurring in the network due to the flow of data packets through the nodes. Collision leads to packet loss. Collision statistics for stationary and mobile environment is compared in Figure.6. Collision for mobile environment is less than stationary environment, because in stationary environment nodes are static, the packet from more than one source reaches the intermediate nodes so that more collision will occur. In mobile environment, nodes are moving at the speed of 20m/s. Collision in mobile environment is less because of the random movement of nodes, packets from source node either reaches the intermediate node or drops the packet in the transmission path. Chance of occurrence of collision is less in mobile environment

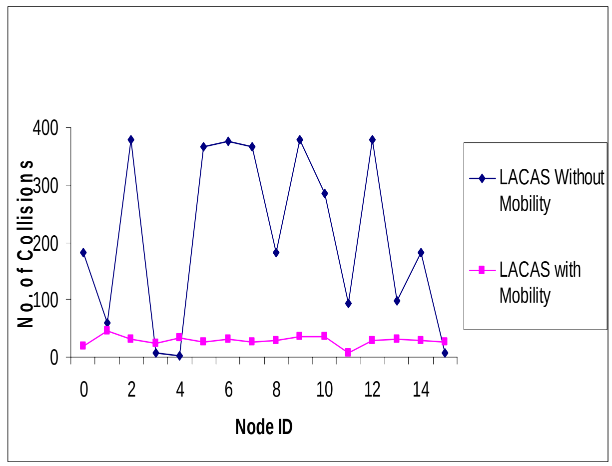


Figure 6 Collision statistics is compared for stationary and Mobile environment

6. CONCLUSIONS

One of the major challenge in WSN is to curb down the congestion. LACAS avoids congestion in effective manner and suits well for stationary environment and shows pronounced performance. But mobile WSN gives feasible healthcare for rural communities and people who can't go for fixed-site clinics. Congestion avoidance for mobile healthcare WSN is addressed by implementing LACAS and its performance is analysed in terms of throughput, energy, and collision in the network. The obtained throughput of LACAS for mobile environment is about 60 percent. Energy consumption in mobile environment is less compared to stationary environment because of no predefined path, packets coming towards the particular node only move to on-state other nodes moves to sleep mode. Collisions statistics in mobile environment is less, due to mobility either the packet will reach the node or lose the packets.

Volume: 02, Issue: 03, Pages: 693-698 (2010)

REFERENCES

[1] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S.Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in-home healthcare: Potential and challenges," in Proc. High Confidence Medical Device Software Systems (HCMDSS) Workshop, June 2005.

[2] C. Y. Wan, S. B. Eisenman and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks", Proceedings of ACM Sensor Systems, California, USA, pp. 266-279, November, 2003.

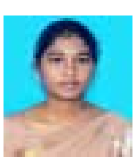
[3] S. Misra, V. Tiwari, and M. S. Obaidat, "LACAS :Learning Automata-based Congestion Avoidance Scheme for healthcare wireless sensor networks", IEEE Journal on Selected Areas in Communication, Vol. 27, No. 4, pp. 466-479, May, 2009.

[4] K. S. Narendra and M. A. L. Thathachar, "Learning Automata - An Introduction". Prentice Hall, 1989.

[5] M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis, "Learning automata: Theory, paradigms and applications", IEEE Transaction systems, man Cybernetics,B, Vol.32,No.6, pp.706-709, December 2002

Authors Biography

Dr. R.Gunasundari received her Ph.D degree in Information and Communication Engineering at Anna University, Chennai under the AICTE Quality Improvement Program in 2009. She completed B.Tech degree in 1995 and M.Tech degree in 1998 in Electronics and communication engineering from Pondicherry Engineering College, Pondicherry. She has 15 years of teaching and research experience. Presently she is working as Assistant Professor in the Department of Electronics and Communication engineering, Pondicherry Engineering College, Pondicherry. Before joining as faculty in Pondicherry Engineering College, she served as Scientist Engineer – C at ISRO satellite center, Bangalore. She has published 7 papers in reputed international journals and presented more than 35 papers in National and International conferences. Her areas of interest include mobile computing, sensor networks and wireless communication networks. She has delivered many guest lectures in various institutions in Mobile Computing, Pervasive Computing, Fault tolerant networks, Mobile IP, Micro Mobility Protocols. She has organized three Workshops on 'Network simulator – ns-2(2) and Familiarization in ns-2 and Glomosim.



R. Arthi received her B.E degree from Madras University in 2002 and M.E. degree in Process control and Instrumentation 2004. She has six years of teaching experience. She is working as Assistant professor in the Department of Electronics and Instrumentation Engineering, St.Josephs College of Engineering, Chennai. She has presented and published more than 5 papers in National/ International Journals and Conferences. Her areas of interest include smart sensors, control engineering, fuzzy logic and artificial Intelligence.

S.Priya received her B.E degree from Annamalai University in 2008 and M.Tech degree in 2010 in Electronics and Communication Engineering from Pondicherry Engineering College, Pondicherry.