LACAS: Learning Automata-Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks

Sudip Misra, Vivek Tiwari, Mohammad S. Obaidat[1], Fellow, IEEE

Abstract: One of the major challenges in wireless sensor network (WSN) research is to curb down congestion in the network's traffic, without compromising with the energy of the sensor nodes. Congestion affects the continuous flow of data, loss of information, delay in the arrival of data to the destination and unwanted consumption of significant amount of the very limited amount of energy in the nodes. Obviously, in healthcare WSN applications, particularly in the ones that cater to medical emergencies or in the ones that closely monitor critically ailing patients, it is desirable in the first place to avoid congestion from occurring and even if it occurs, to reduce the loss of data due to congestion. In this work, we address the problem of congestion in the nodes of healthcare WSN using learning automata (LA)-based approach. Our 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. We maintain that the proposed algorithm, named as *Learning Automata-Based Congestion Avoidance Algorithm in Sensor Networks* (*LACAS*), can counter the congestion problem in healthcare WSNs effectively. An important feature of LACAS is that it intelligently "learns" from the past and improves its performance significantly as time progresses. Our proposed LA-based model was evaluated using simulations representing healthcare WSNs. The results obtained through the experiments with respect to performance criteria having important implications in the healthcare domain. For example, the number of collisions, the energy consumption at the nodes, the network throughput, the number of unicast packets delivered, the number of packets delivered to each node, the signals received and forwarded to the Medium Access Control (MAC) layer, and the change in energy consumption with variation in transmission range, have shown that the proposed algorithm is capable of successfully avoiding congestion in typical healthcare WSNs requiring a reliable congestion control mechanism.

---

[1] Sudip Misra is with the School of Information Technology, Indian Institute of Technology, Kharagpur, India. He can be contacted by e-mail at sudip_misra@yahoo.com.
Vivek Tiwari is with the Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur, India. He can be contacted by e-mail at ece.vivek@gmail.com.
Mohammad S. Obaidat is with the Department of Computer Science, Monmouth University, New Jersey, USA. He is a *Fellow of the IEEE* and *Fellow of the SCS*. He can be contacted by e-mail at obaidat@monmouth.edu.

Keywords: Wireless sensor networks (WSNs), congestion control, learning automata, healthcare applications, performance evaluation.

1. Introduction

Due to the growing demand for low cost "networkable" sensors, in conjunction with the recent developments of Micro-Electro Mechanical System (MEMS) and Radio Frequency (RF) technology, new sensors come with advanced functionalities for processing and communication. A WSN consists of such sensors, which have tight constraints with respect to computational power, storage and energy resources [1]. WSNs, in recent years, have advanced in leaps and bounds due to their innumerable applications in various fields including the military, civilian, mining, healthcare and scientific monitoring for commercial purposes. One of the popular application domains of WSNs is healthcare, specifically, the remote monitoring of the conditions of ailing patients [2, 3, 4, 5, 6]. In general, in healthcare applications, the sensor nodes are, typically, deployed over a region in space and, based on the specific healthcare related task they are targeted for, they generate data (such as the pulse rate of a critically sick patient), which are eventually delivered to the control center for analysis by medical personnel. As such, WSNs are characterized by several features, out of which their unique network topology, diverse applications, distinct traffic characteristics and message size are of concern. Additionally, the hard energy consumption constraints imposed on the nodes in WSNs make the different protocols and mechanisms devised for these networks consider the energy consumption parameters with prime importance. Quite intuitively, superposing the healthcare criticalities, such as fast response to medical emergencies, high reliability of transmission of data from the source nodes to the sink node and their prompt delivery, bring further challenges in healthcare WSNs.

In this paper, we focus only on the issue of congestion in healthcare WSNs. In particular, we focus on large-scale medical disaster response applications. As will be discussed elaborately in Section 2.1, congestion is a severe problem in such situations. Obviously, it is unwanted to have packets containing life-critical information being queued-up in the intermediate nodes in the multihop paths thereby delaying the transmission of information, or such packets being dropped because of occurrences of congestion in the intermediate nodes. In short, the issue of occurrences of congestion in WSNs is linked to the following. Congestion leads to either dropping the packets at the intermediate nodes or the formation of queues in them, which, again, in effect, would lead to packet delay. These challenges are to be mitigated in a very effective and efficient manner so that fairness, in terms of the distribution of the packets amongst the nodes, is guaranteed without unwantedly withering away much of the nodes' energy. There are two terminologies relating to how congestion could be handled – one is congestion control and the other is congestion avoidance. While some researchers have used them in the same context, in this paper, in order to avoid confusion of the terminologies, we maintain the following stance, although the preference of choice of the terminologies has minimal impact, if any, on the run through our proposed solution in the rest of the paper. We maintain that while the former attempts to treat the problem once it occurs, the later avoids the occurrence of the problem before its onset. The previous works, in fact, typically, take a curative approach to deal with congestion, rather than a preventive one. Using automata stationed in the sensor nodes, our algorithm "intelligently, attempts to avoid the onset of congestion. Having said that, obviously, the complete eradication of the onset of congestion is difficult. Our approach minimizes the chances of occurrences of congestion to the extent possible, while saving some of the important/critical network resources.

1.1. Motivation

Although in this work we specifically targeted healthcare WSN applications, the problem of congestion is not limited to these. The problem of congestion exists almost in all types of networks. Without specifically citing the different network types and the congestion control and avoidance schemes that exist in them, it suffices to mention that congestion is probably one of the most important concerns against affirming the reliability of transmission of information in any network – this is especially true for healthcare WSN applications due to the time criticality and content criticality of data carried in them. In the case of WSNs, congestion in the network increases traffic to such an extent that energy is dissipated in a colossal amount in the sensor nodes and also it leads to loss of packets, thereby creating a barrier for fair and reliable flow of packets. In many healthcare applications of WSNs (such as sensors installed inside the body of humans undergoing internal organ rehabilitation of some kind) [4, 5, 6], the nodes are mandated to work uninterruptedly for months and, possibly, years together in a continuum, without having the possibility of replacing the sources of energy in them. Therefore, optimizing the consumption of energy at the sensor nodes becomes an issue of prime concern. Typically, in most healthcare applications, and especially in the case of the medical disaster relief WSN applications, different nodes sense a large number of events. All these nodes, then, try to transmit the information to the sink node (control centre) with the help of other intermediate nodes and localized control centers, if any. This leads to increased possibilities of congestion at the intermediate nodes. Coupling this fact with the high levels of energy limitations in the sensor nodes makes the congestion problem in healthcare WSNs, evidently, more challenging than the congestion problem in other application domains. It is this challenge that has inspired us to address this problem in this paper.

Perhaps the most fundamental step that should be taken in avoiding congestion is limiting the flow of packets at the intermediate nodes to an apt value so that smooth, fair and reliable packet flow takes place in the network. This is what we did in this work using an LA-based approach.

The idea of using LA to address the congestion control problem in healthcare WSNs is novel. As a matter of fact, we are unaware of any existing LA-based approach that addresses the congestion control problem in any other type of network. Our LA-based approach is designed in such a manner that the automata stationed in the intermediate nodes of the network continuously interact with the environment and adaptively learns, depending on the traffic load at each node, the optimum rate at which the rate of flow of data should be maintained. This rate can change, when the network traffic congestion level changes. Essentially, this restricts at all times the flow of data through a node to an optimum level and, in effect, prompts other nodes through other multihop paths to relay the data.

LA has found a number of applications for solving different engineering problems (e.g., [14]-[31]. Its attractiveness lies in its ability to successfully address complex engineering problems characterized by high levels of uncertainty. The applications of LA are not myopic to any single domain. LA can help to adaptively learn optimum actions amongst different candidate actions offered to an automaton. Our LA-based approach limits the number of packets flown through the intermediate nodes so that a node does not get overloaded by the number of packets and eventually leads to a congestion-free and energy-efficient network system. Our prime motive was to increase the efficiency of the network and this was, in turn, motivated by the need for better utilization of some of the critical network resources for healthcare services.

2. Related Works:

In recent years, a number of research efforts worldwide have targeted how the congestion problem in sensor networks can be countered (e.g., [7, 8, 9, 10, 11, 12, 13]). Before elaborating on our work and the specific contributions we have made, and to help the readers understand the typical approaches that exist currently, let us review briefly a few of the *popular* congestion control mechanisms that exist currently.

One of the popular congestion control mechanisms for WSNs that is also based on adaptation to data flow characteristics, like ours, is adaptive rate control (ARC) [7]. 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 $0<\beta<1$.

In another popular algorithm, CODA [8], congestion is detected based on the queue length of packets at the intermediate nodes. CODA comprises of three elements – 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 multi-source regulation, the source gets an ACK from the sink and when the congestion occurs, the sink stops sending ACKs to the source. Like ARC, CODA also controls the rate of flow of packets based on the AIMD algorithm. This technique is energy-efficient, but the reliability of the successful delivery of the packets to the destination is not guaranteed because, on receiving backpressure signals, meaning that the signals received by the source node from the intermediate nodes, the nodes drop their packets based on the congestion parameters.

Some other congestion control protocols such as Fusion [11], PCCP [9] (priority based congestion control protocol), CCF [10] (congestion control and fairness), Trickle [12] and Siphon [13] have also been proposed. Fusion controls the congestion in stop and start manner. Here the neighboring nodes stop emitting data packets when congestion is detected in the network. PCCP also uses rate adjusting algorithm unlike that of the AIMD technique. CCF controls the congestion based on the packet service time by adjusting the transmission rate. Trickle achieves performance comparable to TCP/IP. It has a periodical event in each node that suppresses broadcasting if the metadata that it receives from its neighboring node exceeds the threshold. Siphon uses virtual sinks in the network to remove data events from the sensor network when any symptom of traffic load occurs.

We planned to mitigate the congestion problem 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. Most of the existing protocols (discussed earlier), typically, approach the congestion problem by attempting to constantly change the rate of the flow of packets at the source node, based on congestion notifications from the intermediate nodes. This can lead to delays in the delivery of the packets, which is undesirable for the medical disaster applications, for reasons stated earlier in our discussions. What we attempted to do is that, instead of changing the data flow rate at the source at all instances of occurrence of congestion, the intermediate nodes themselves can be fed with the knowledge of the traffic flow characteristics at all instances so that congestion can be avoided even before it occurs.

To be more specific, the *contributions* of this work can be summarized as follow:

- We have designed a LA-based scheme for healthcare WSNs, which avoids congestion in a very effective manner. All the intermediate nodes have automata stationed in them, which are tasked to monitor and control the rate of flow of data through them, so as to minimize the likelihood of occurrences of congestion.
- We base our approach on equating the packet arrival rate and the packet service rate, i.e., we try to make both of these rates equal, preventing any kind of queuing at the nodes to a large extent and, hence, any packet delay because of that. Each automaton is tasked to compute the "best" or, to be precise, the most optimal rate of flow of packets, based on the number of packets dropped and, finally, that action is selected to be the output of the system comprising of the automaton adaptively interacting with its environment.
- Our proposed algorithm is capable of adaptively learning and "intelligently" choosing "better" data rates in the future, based on the past experience with congestion with the other data rates.
- Our proposed algorithm does not require the source nodes to be fed back by the intermediate nodes to slow down, as is, typically, done in many existing WSN congestion control schemes. In our approach, we save this time, by making the intermediate nodes to take a proactive approach in controlling the current rate of flow of packets, thereby improving the overall performance of the network.
- Our approach is useful in large-scale WSN healthcare applications having increased likelihood of congestion for reasons that are elaborated in Section 2.1.

2.1. Healthcare WSN and Congestion Avoidance

The popularity of WSNs being used in healthcare is attributed to the properties of WSNs such as reliability, interoperability, efficiency, wearability, low-power consumption and inexpensiveness. Continuous patient monitoring and diagnostics [23], most often remotely, by doctors and nurses, without them being physically present in the patient sites is one of the popular applications of WSNs today. Figure 1 shows how a WSN can be deployed for remote patient monitoring. Sensors are attached to different patients and those sensors are capable of sensing vital patient information that can be transmitted to the control center with the help of some other neighboring nodes. Although, as shown in Figure 4, some of the WSN nodes (e.g., a sensor attached to a patient in an ambulance) in such healthcare applications, can be mobile, in our current work, we have restricted ourselves to such healthcare applications in which all the sensor nodes are stationary (at least they do not change their locations for a few hours), e.g., monitoring a number of patients donating vital organs in a organ donation camp).

Irrespective of the specific patient monitoring application in which the WSNs are used, the events sensed and the information collected by the different sensors about different patients, are transferred to the control centre, typically, through a multihop path, for proper and continuous monitoring. Let us consider the times of large-scale disasters, whether caused due to any natural calamity such as occurrences of earthquakes and spread of epidemic diseases, or due to human acts, such as the 9/11 terrorist attacks on the World Trade Center in New York, or mass casualties in a battlefield during a war, the doctors in hospitals and relief camps need to monitor hundreds of patients at the same time. Often, the number of doctors designated to look after the patients is much lesser compared to the number of patients needing attention. In such cases, use of WSNs can be indispensable for both pre-hospital and in-hospital patient monitoring. For example, after the patients are admitted to the hospital in bulk, after first aid treatment, or even prior to that, the patients can be equipped with specialized sensor devices capable of monitoring vital patient information such as the heart rate, breathing condition and possibilities of formation of blood-clots.

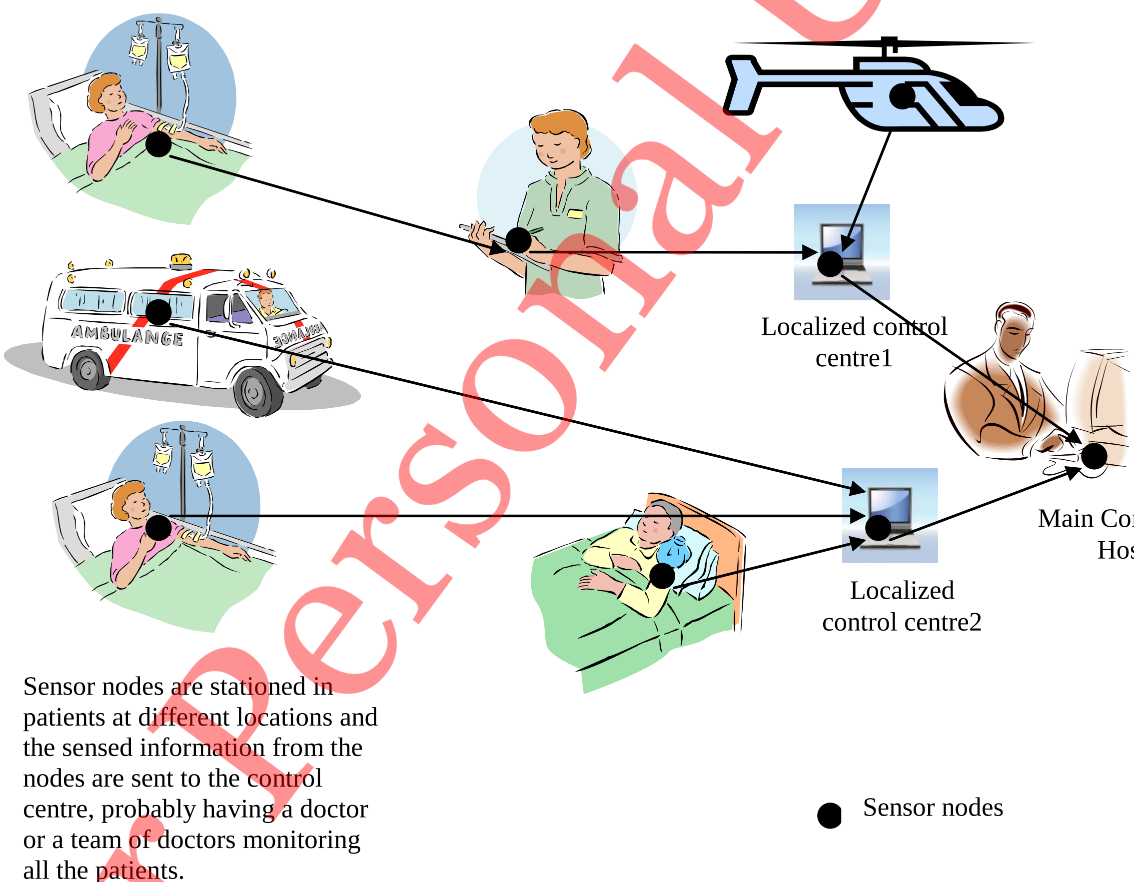

Figure 1: A Sensor Network for a Large-Scale Medical Emergency Situation Involving a Number of Patients Being Monitored at the Hospital Control Room

*Challenges/Issues***:** Although the above applications of WSNs, for example, for remote patient monitoring during large-scale medical emergencies appear to be potentially appealing, there exists an associated challenge – the challenge of congestion control – that is typical to such applications[2]. During such large-scale medical emergencies, it is quite likely that the sensors placed in the different patients, will sense and transmit vital patient information very frequently and simultaneously, leading to increased likelihood of congestion in the networks in such applications than in other healthcare WSNs. As congestion in WSNs leads to dropping of packets at the nodes, increased consumption of limited energy in the nodes and reduction of the throughput of the network. Evidently, in such life-critical applications involving a large number of patients, congestion is extremely undesirable. The implications of congestion in such situations can be appraised better if we consider a case in which the packets carrying information of a dying patient gets dropped due to congestion in a node. Such an event can be so disastrous that it can lead to the death of a patient, which would have, otherwise, survived, if those packets reached their destinations on time. Obviously, it is unlikely that the occurrences of congestion can be eliminated completely. However, what can be definitely achieved is significantly reducing the effects of congestion, i.e., significantly decreasing the number of packets that gets dropped due to congestion, the large amount of unwanted consumption of the limited energy at the sensors and increasing the number of packets that get successfully delivered with respect to the number of packets that are sent from the different nodes.

We addressed this problem by proposing a novel approach using which congestion can be avoided effectively. Through simulations, we establish in Section 6 that our approach is indeed capable of significantly reducing the number of packets that can get dropped due to congestion, the amount of energy that gets consumed at the nodes due to congestion and increasing the throughput of the network. Additionally, we should mention that, obviously, the back-buffering approach, in which the source node sending the data is fed back by the intermediate nodes to slow down, which is typically used in the previous schemes such as CODA [8], is of limited help in such life-critical applications requiring low latency. As mentioned earlier, we attempted to address this problem by taking a preventive approach rather than a curative one. In our solution mode, we station an autonomous learning machine (automaton) at the sensor nodes. These automata can be perceived to be small pieces of code that can interact with the environment and make intelligent decisions based on the environment characteristics. In our approach, we make these automata to act from the start itself and adjust to a data flow rate that significantly minimizes the likelihood of the dropping of packets and the consumption of energy at the nodes and maximizes the

[2] Although congestion is more likely to occur frequently in medical emergencies and disaster situations, other healthcare WSNs (such as where a sensor network is used to monitor the movement of the limbs of a newborn) may also relatively infrequently experience different levels of congestion.

throughput of the network. In sum, our approach helps us to avoid congestion in a "busy" network, so that a continuous and non-interruptible monitoring and diagnosis of the patients can take place without losing vital patient data.

3. Congestion Problem for Many-to-One Traffic Pattern in Healthcare WSNs

For congestion control in WSNs, for most healthcare applications, we primarily deal with many-to-one traffic patterns. In many-to-one traffic patterns, we have a single sink node and multiple source nodes which can be considered to be affixed with the patients. The congestion takes place when the sink node (control centers) in the network does not find enough resources, for the data packets, received from the multiple sources.

3.1. Congestion Control in WSN

The congestion control problem is primarily localized to the MAC layer (Data-Link layer), as shown in Figure 2[3]. Typically, most existing congestion control efforts limit the flow of packets through a node in a suitable manner – the exact approach depends on the specific mechanism that is used.

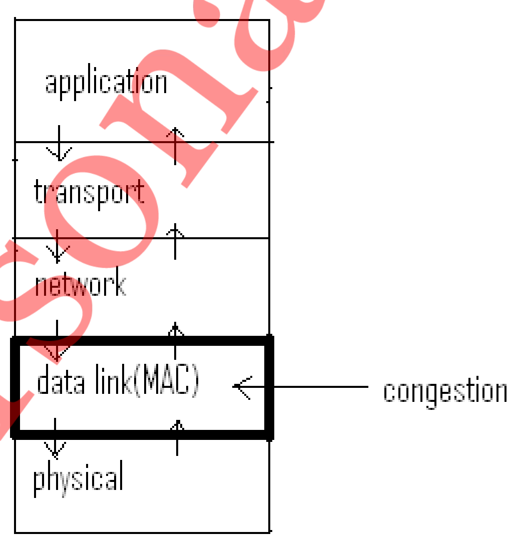

Figure 2: Network stack model

In typical WSNs, a source node emits data packets at some specific CBR. These data packets reach the intermediate nodes in the multihop path to the sink and the intermediate nodes, in turn, forward the

[3] In Figure 2, we show only the five important layers that are used. Since the layers, other than MAC, are not much concern to us in this work, for the sake of brevity, we do not elaborate on them here. However, interested readers may refer to standard books on WSN such as [1].

packets to the next hop node. This process continues until the packets reach the final destination. This is diagrammatically shown in Figure 3. If, in an intermediate node, the rate at which the node receives the packets is not equal to the rate with which it is forwards it to the next node, different problems such as dropping of packets, queuing delay, collisions and congestion arises. Also, the problem is prominent due to the many-to-one traffic or funnel pattern [24] of WSN, wherein the data is sent from multiple sensor nodes to a destination continuously with the help of other intermediate nodes in the multihop path.

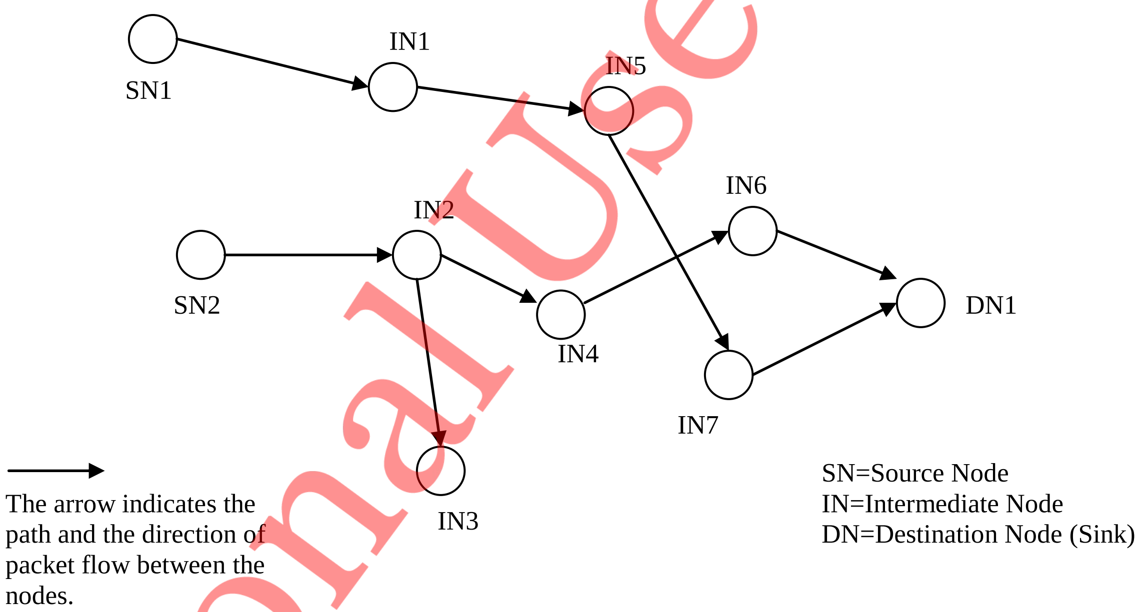

Figure 3: Basic traffic pattern in WSN

3.2. Rate Control

As controlling the rate of flow of traffic to a sensor node is one of the generic approaches for congestion control, few existing works discuss mechanisms for doing it. Examples include source control mechanism and hop-by-hop backpressure mechanism [7, 8]. In sum, in the source control mechanism, the rate of flow of data is carried out at the sink node or the destination node. On detection of congestion, the sink informs the source to control its rate of dispatch of packets. In hop-by-hop backpressure mechanism, the intermediate nodes, based on their congestion, state informs the source to adjust their rate.

In our approach, we controlled the rate of flow of data in a fashion that includes the intermediate nodes as well as the sink node. In our approach, the intermediate nodes do not provide any feedback to the source node; rather, the intermediate nodes adjust themselves according to the output of automaton stationed in them. The automaton does that by continuously interacting with the environment. This helps the node to select the rate of transmission of packets. Compared to the existing source control and the hop-by-hop backpressure mechanisms mentioned earlier, our approach helps us to increase the efficiency of the network by optimizing the unnecessary consumption of valuable network resources such as the residual energy at the nodes and the link capacities. In Section 4, we discuss our approach in detail.

4. LACAS: Our LA-Based Approach for Congestion Avoidance in WSN

In this Section, we elaborate on the LA-based approach that we took to avoid congestion in WSNs. Before describing our solution approach in Section 4.2, we first introduce in Section 4.1 the concepts underlying the theory of LA. These concepts are essential for understanding the different elements constituting our proposed congestion-avoidance algorithm.

4.1 LA

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 latter types of automata are of interest to the research results we report in this paper, and are termed as *adaptive automata*. The term "*learning*" refers to the act of acquiring knowledge and modifying one's behavior based on the experience gained. Thus, in our case, the adaptive automata we study in this paper, adapt to the responses from the Environment through a series of interactions within them. The automata, then, attempt to learn the best action from a set of possible actions that are offered to them by the random stationary or non-stationary environment in which they operate. The automata, thus, act as decision makers to arrive at the best action.

The operation of LA can be best described through the words of the pioneers Narendra and Thathachar [25]: "… a decision maker operates in the random environment and updates its strategy for choosing actions on the basis of the elicited response. The decision maker, in such a feedback configuration of decision maker (or automaton) and environment, is referred to as the *learning automaton*. The automaton has a finite set of actions, and corresponding to each action, the response of the environment can be either favorable or unfavorable with a certain probability" ([25], pp. 3).

LA finds applications in optimization problems in which an optimal action needs to be determined from a set of actions. It should be noted that in this context, learning might be of best help only when there are high levels of *uncertainty* in the system in which the automaton operates. In systems with low levels of uncertainty, LA-based learning may not be a suitable tool of choice [25].

A comprehensive overview of research in the field of LA can be found in the classic text by Narendra and Thathachar [25], and in the August 2002 special issue of the *IEEE Transactions on Systems, Man, and Cybernetics, Part B [28]*. However, to ease out the understanding of the philosophy underlying our solution approach, we briefly review below some of the fundamental concepts.

4.1.1 The Automaton

The Automaton, in our case, is, generically defined by a quintuple {A, B, Q, F(.,.), G(.)}, where [25]:

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

(ii) 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.

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

(iv) 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.

(v) 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 [25, 28]. Without loss of generality, G is deterministic.

4.1.2 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 [25].

(i) $A= \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents a finite input set

(ii) $B= \{\beta_1, \beta_2, \ldots, \beta_l\}$ is the output set of the environment, and

(iii) $C= \{c_1, c_2, \ldots, c_r\}$ 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 4. 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 $\{\alpha_1, ..., \alpha_r\}$ 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.

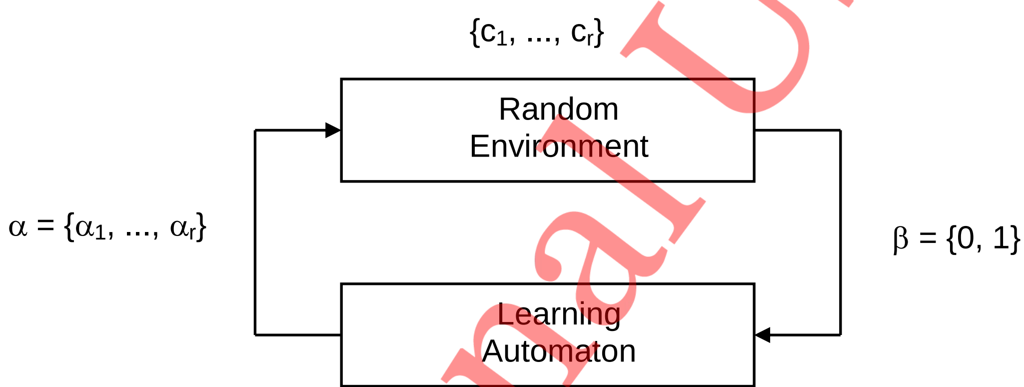

Figure 4: The Automaton-Environment Feedback Loop [25, 28].

We now provide a few important definitions used in the field of LA. Given an action probability vector $\mathbf{P}(t)$ at time 't', the *average penalty* is defined as [25]:

$$\begin{aligned} M(t) &= E\left[\beta(t)|P(t)\right] = \Pr\left[\beta(t)=1|P(t)\right] \\ &= \sum_{i=1}^{r} \Pr\left[\beta(t)=1|\alpha(t)=\alpha_i\right] \times \Pr\left[\alpha(t)=\alpha_i\right] \\ &= \sum_{i=1}^{r} c_i p_i(t). \end{aligned} \tag{1}$$

The average penalty for the "pure-chance" automaton is given by:

$$M_0 = \frac{1}{r}\sum_{i=1}^{r} c_i. \tag{2}$$

As $t \rightarrow \infty$, if the average penalty $M(t)<M_0$, at least asymptotically, the automaton is generally considered to be better than the pure-chance automaton. E[M(t)] is given by:

$$E\left[M(t)\right]=E\left\{E\left[\beta(t)|P(t)\right]\right\}=E\left[\beta(t)\right]. \qquad (3)$$

4.1.3. Action Probability Updating

In our work, we deal with the *Variable Structure Stochastic Automata* (*VSSA*). 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 (L_{RI}) scheme. It is based on the principle that whenever the automaton receives a favorable response (i.e., reward) from the environment, the action probabilities are updated, whereas if the automaton receives an unfavorable response (i.e., penalty) from the environment, the action probabilities are unaltered. Mathematically, it is presented below [25, 28]:

$$\begin{aligned}
&p_i(n+1) = 1 - \Sigma_{j\neq i}\, \lambda_r\, p_j(n) && \text{if } \alpha_i \text{ is chosen and } \beta = 0 \\
&p_j(n+1) = \lambda_r\, p_j(n) && \text{if } \alpha_i \text{ is chosen and } \beta = 0 \qquad (4)\\
&p_j(n+1) = p_j(n) && \text{if } \alpha_i, \alpha_j \text{ chosen, and } \beta = 1,
\end{aligned}$$

where λ_r $(0 < \lambda_r < 1)$ is the parameter of the scheme. Typically, λ_r is chosen to be close to unity. Note that only rewards are processed in this scheme. Therefore, if α_i is chosen and it receives a reward, the probability of choosing this action in the next iteration, $p_i(n+1)$, must be increased. This is accomplished in two steps. First, the probabilities of choosing any other action α_j, for all $j\neq i$, on the next iteration are reduced by setting $p_j(n+1)$ to $\lambda_r\, p_j(n)$ for all $j\neq i$. Next, the probability of choosing α_i on the next iteration, $p_i(n+1)$, is increased by subtracting the sum of all $p_j(n+1)$ for $j\neq i$, from unity.

5. LACAS

Let us now explain the specifics of our LA-based model, LACAS, which can be used for congestion control in WSNs. An automaton, which we explained earlier, to be a simple autonomous machine (code) capable of making decisions, is stationed at every node in the network, as shown in Figure 5.

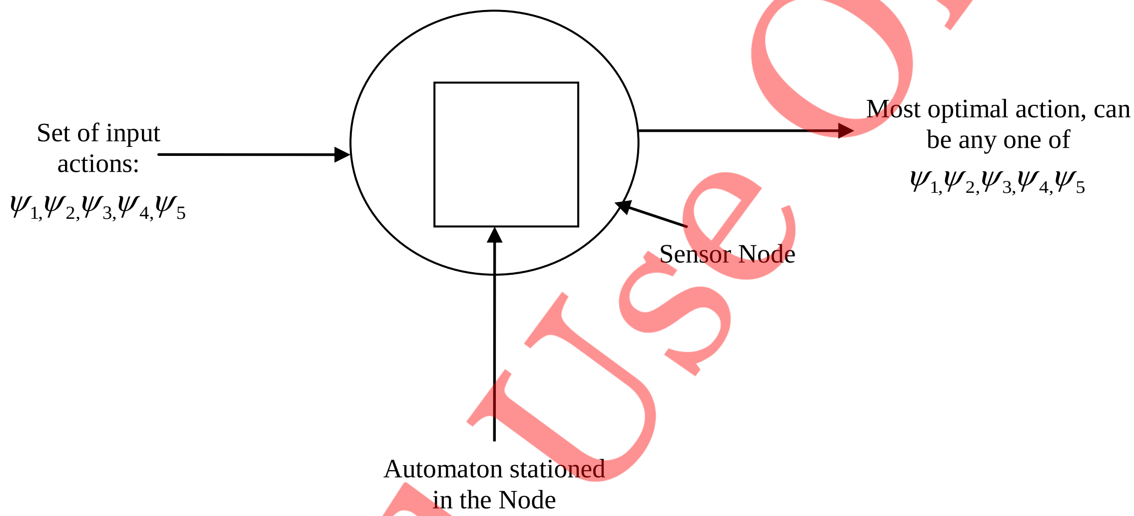

Figure 5: Diagram showing a node with an automaton stationed in it.

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, if we 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, we should 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.

For the input to the automaton at time, t=0, in our work, we limited the number of actions associated with an automaton to 5 [4], which are based on the rate with which an intermediate sensor node receives the packets from the source node. We denote these actions by $\psi = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5\}$, as shown in Figure 5.

[4] However, we should clarify that, although our algorithm is explained and the results presented in this paper are with the help of 5 actions, the choice of the number of actions is purely arbitrary. It is designed to be a parameter that can be set by the user. It can be, trivially, established through theoretical means, that the choice of the number of actions does not affect the overall functionalities of the algorithm.

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, ψ_1, and rewards/penalizes that action based on the packets dropped at the node. If the action, ψ_1, is rewarded, the probability value of ψ_1 is increased and the probability values of the other actions, i.e., $\psi_2, \psi_3, \psi_4, \psi_5$, are decreased as per the equations shown below in (5a) and (5b)

$$P_{\psi_1}(n+1) = P_{\psi_1}(n) + \frac{1}{\lambda}(1 - P_{\psi_1}(n)) \qquad (5a)$$

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

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[5]. This is shown, mathematically, in Equations (6a) and (6b).

$$P_{\psi_1}(n+1) = P_{\psi_1}(n) \qquad (6a)$$

$$P_{\psi_{2,3,4,5}}(n+1) = P_{\psi_{2,3,4,5}}(n) \qquad (6b)$$

The probability values associated with all the actions should be such that at every time instant the summation of their probabilities should equal unity, i.e., $\sum_{i=1}^{r} P_{\psi_i}(n) = 1$. 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

[5] In our approach, we follow the L_{RI} probability updating scheme, which is known to be efficient [25, 28].

cycle. This vicious cycle will continue until the most optimal action is selected, i.e., $\lim_{t\to\infty} P_{\psi_i}(n) = 1$, 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 [6]. The above-mentioned approach is also followed at all the nodes of the network.

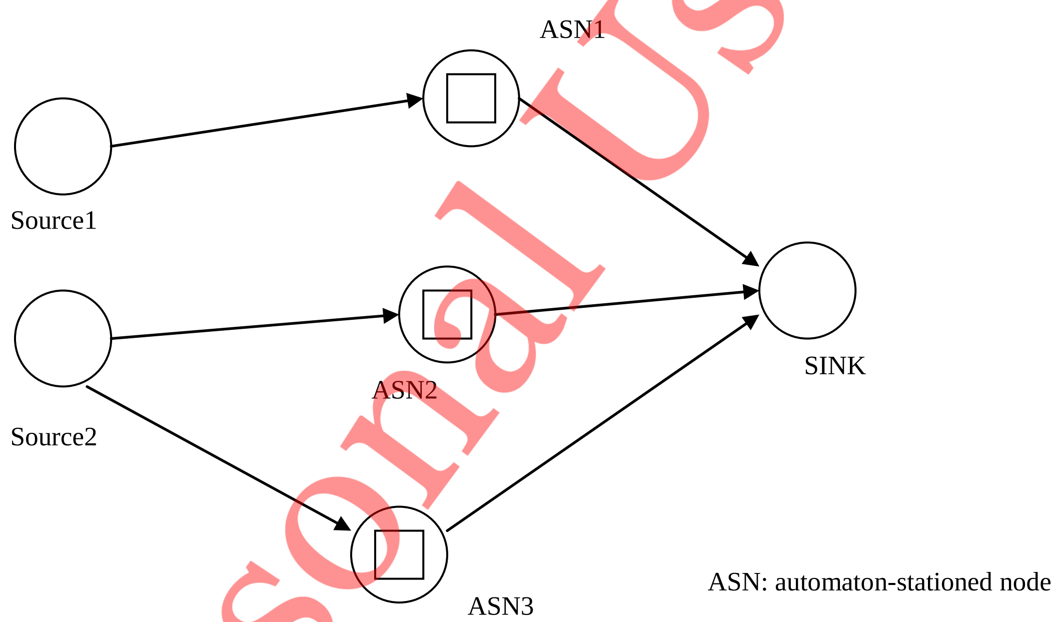

Figure 6: The figure demonstrates the network model consisting of 7 nodes. Also, it shows that the automata stationed in the intermediate nodes help in the adaptation of the data flow rate.

We, now, present the important steps of the LACAS algorithm. This algorithm executes at every intermediate node in the network that has an automaton stationed in it.

[6] We should clarify that we are considering a stationary environment in which the probability distribution itself does not change with time. It is worth restating that, if such is not the case, the algorithm will not function as explained in the paper. The LA-based analysis of problems involving non-stationary distributions is non-trivial. We plan to consider non-stationary distributions as a work to be done in the future.

Algorithm: LACAS

Input:

- G: The network.
- N': The number of nodes in G.
- N: The number of nodes acting as intermediate nodes during any run.
- Set of actions $\psi = \psi_{1,} \psi_{2,} \psi_{3,} \psi_{4}, ..., \psi_{n}$ for an intermediate node.
- Rate corresponding to each of the actions $\psi_i = \omega_i$ bits/sec in an automaton, where $1 \le i \le n$.

Output:

- For each automaton, the rate corresponding to the most optimal action ψ_o, where $1 \le o \le n$.

Principal Steps:

BEGIN

Step 1: /*********__Put the loop for all the intermediate nodes__********/

For every intermediate node, execute Steps 2 to 4 until $\lim_{t \to \infty} \mathrm{P}[\psi_i] \to 1$..

Step 2: /*******__Initialize the probability of selecting an action from the set of actions__ $\psi_{1,} \psi_{2,} \psi_{3,} \psi_{4}, ..., \psi_{n}$ *******/

Initialize the probability of selecting $\psi_i = P_{\psi_i}$.

Step 3: /*********__Select an action randomly out of n actions present__ ********/

Choose an action randomly out of a set of *n* actions as follows:

ψ_i =rand ( )%n+1.

Step 4: /*********__Update the probabilities at every node until an optimal action is chosen at that node__ *********/

while(1)

{

if action ψ_i is chosen and the Environment response, $\beta = 0$, update the probabilities according the following scheme:

$$P_{\psi_i}(n+1) = P_{\psi_i}(n) + \frac{1}{\lambda}(1 - P_{\psi_i}(n))$$

$$P_{\psi_j}(n+1) = (1 - \frac{1}{\lambda}) P_{\psi_j}(n)$$

else update the probabilities according to the following scheme:

$$P_{\psi_i}(n+1) = P_{\psi_i}(n)$$

$$P_{\psi_j}(n+1) = P_{\psi_j}(n)$$

}

Step 5: /*****__Transmit the packets corresponding to the rate selected by the automaton__********/

Transmit data packet with the rate ω_i bits/sec corresponding to ψ_i

END

Example: To help the readers better understand our solution approach, let us consider the small sample network topology that consists of six nodes, as shown in Figure 6. In this example, two nodes are shown as source nodes, three nodes as intermediate nodes in which automata are stationed and one node as a sink node. Let us consider the automaton stationed in the intermediate node, denoted by ASN2. The automaton in ASN2 is provided with a set of inputs as action ψ . As required in Step 2, each of the actions of ASN2 is initialized. After getting the actions as the input, the automaton selects an action ψ_i randomly out of the available 'N' actions, as per Step 3 of LACAS. The selected action's probability, P_{ψ_i}, will be updated, based on whether it is rewarded or penalized by the environment, as shown in Step 4. The process of updating the probability will go on until an action, ψ_i , is selected whose probability tends to unity at time t tending to infinity. The above mentioned steps will occur even for the nodes, ASN1 and ASN3, since an automaton is stationed in them too. Finally, after a suitable action is selected by the automaton, the node will transmit data packets corresponding to the rate of selected action ψ_i as stated in Step 5.

6. Simulation Results

We evaluated the performance of LACAS through simulations. Simulation studies were performed using the Global Mobile Simulator (GloMoSim) [26, 32]. We compared the performance of LACAS by simulating it in both the congested and the normal modes of the network. The results of simulation show that LACAS, elaborated below, that LACAS is, indeed, capable of curbing down congestion in the nodes of the network significantly. In this Section, we will first present the configurations of the simulator which were used in our studies. Following this, we will elaborately present the results we have obtained.

6.1. Setting and Configuration

The testbed for the simulation consisted of 100 nodes. In the *congested mode*, we selected 5 sources (Node 1, Node2, Node3, Node70, and Node80) and 4 sinks (Node4, Node6, Node99, Node100) and the number of packets to be emitted was set to 10,000, each packet having a size of 2,048 bytes. The rest of the nodes were set as intermediate nodes. In the *normal mode*, the number of packets was reduced to 1,000, but the sizes of the packets were kept the same. Also, the nodes used as sources and sinks were kept unchanged. The LACAS algorithm was evaluated in the congested mode to check its effectiveness in reducing congestion. The LACAS implementation was done by modifying the 802.11 protocol of the MAC layer. There were other choices of MAC layer protocols such as CSMA, MACA, TSMA [1, 27], but we decided to restrict ourselves to 802.11, because it effectively deals with the problem of hidden node and exposed terminal problems [1, 27, 33]. A summary of the simulation parameters set in our experiments is presented below.

Simulation Time	: 100 secs.
Seed for generation of random numbers	: 3
Terrain Dimension	: (500m, 500m)
Number of Nodes	: 100
Node Placement	: Uniform/Grid/Random
Mobility	: None
Propagation limit	: -111.0 dBm
Propagation path loss	: Two ray
Noise figure	: 10.0
Temperature	: 290.0 Kelvin
Radio Type	: Radio-Accnoise(standard radio model)
Radio Rx type	: SNR bound
Radio Tx power	: 10 dBm/15dBm/20dBm
MAC protocol	: 802.11
Routing	: Bellman-Ford Algorithm

6.2. Performance Metrics

Various performance metrics have been considered for evaluating the performance of LACAS. The main measures of performance were:

- *Energy consumption*: Sensor nodes, especially, those used in healthcare applications, have tight constraints in terms of energy. Any protocol designed specifically for healthcare WSNs should take the amount of energy consumption into account. The criticality of the amount of consumption of energy can be understood better by considering the following examples related to remote patient monitoring. Let us consider an application in which the activities of the vital organs (e.g., heart, lungs and brain) of a patient undergoing a critical surgery are monitored using a network of sensors. In such a case, it would be extremely undesirable to have one of these sensors run out of battery and stop working. To appreciate the importance of energy consumption in the nodes in healthcare WSN applications, let us cite another example. Referring to the example shown in Figure 1, which shows a network of sensors for monitoring the pre-hospitalized and the post-hospitalized patients during a large-scale disaster, it would be undesirable to have the sensor monitoring a critically injured patient in an ambulance runs out of battery. It is, therefore, required to measure the amount of energy the sensor nodes in LACAS consume. It would be extremely undesirable to have a congestion control/avoidance scheme that consumes a lot of energy.

- *Throughput*: Throughput measures the average rate of a successful packet delivery over a communication channel. It is a measure of the number of packets received with respect to the number of packets that are transmitted. As mentioned earlier, this measure is important in networks for healthcare applications, because the importance of the reliability of the transmission of packets, and the rate in which the packets, especially the ones carrying vital patient/medical data, are received at the destination.
- *Number of collisions*: This metric captures the total number of collisions occurring in the network due to the flow of data packets through the nodes. Collisions result in loss of packets. This is also an important measure of performance for any MAC protocol designed to work in healthcare domain, in which critical patient information is carried.

A few other metrics were also considered in our study. They are self-explanatory and are not elaborated in this Section. The simulation results obtained using all these metrics are presented in Section 6.3.

6.3. Performance Results

Energy Consumption: Figure 6 shows the energy consumption, in mWh, at the different nodes of the network in the congested mode, normal mode and with LACAS.

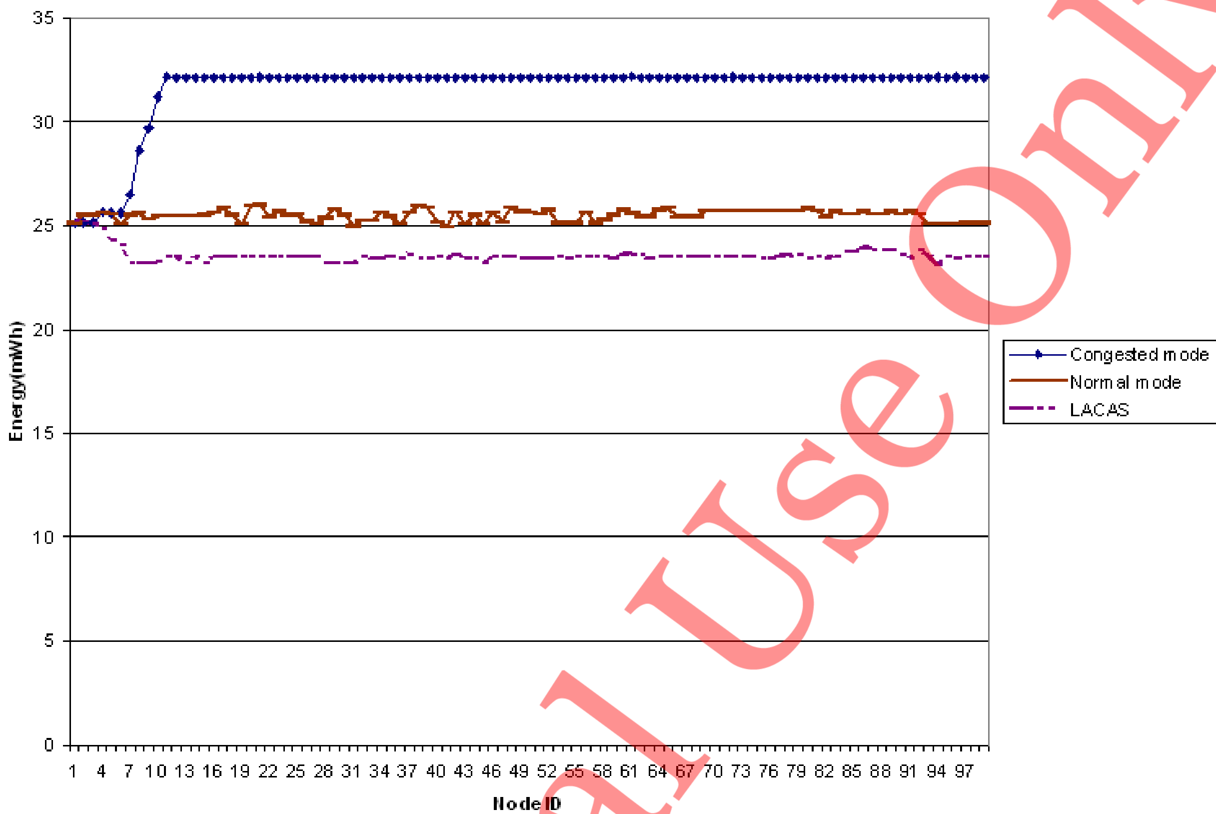

Figure 7: Energy consumption by the Nodes

As is evident from Figure 7, the energy consumed during the transfer of packets is less in the case of LACAS compared to the congested and the normal modes. In the congested mode, the packets arriving from different nodes collide, because the flow of packets is not controlled. In the case of LACAS, the rate of flow of packets is controlled, leading to a fair and reliable transfer of the packets in the network. This leads to reduced energy consumption at the nodes compared to the congested and the normal modes. This is an important achievement of LACAS, because, as mentioned earlier, the nodes in WSNs are extremely energy constrained and in most applications it is infeasible to replaces the sources of energy in the nodes, when they run out of power. Another observation of importance is that, initially, the energy consumption at the nodes is not significantly different. However, as the simulation includes more and more nodes into the network, the energy utilization also increases. However, in LACAS, throughout the network, all of the nodes have similar levels of energy consumption, which is better than that in the congested and normal modes.

Collisions: The comparative results of the number of collisions taking place in the network in the normal mode, the congested mode and with LACAS are shown in Figure 8.

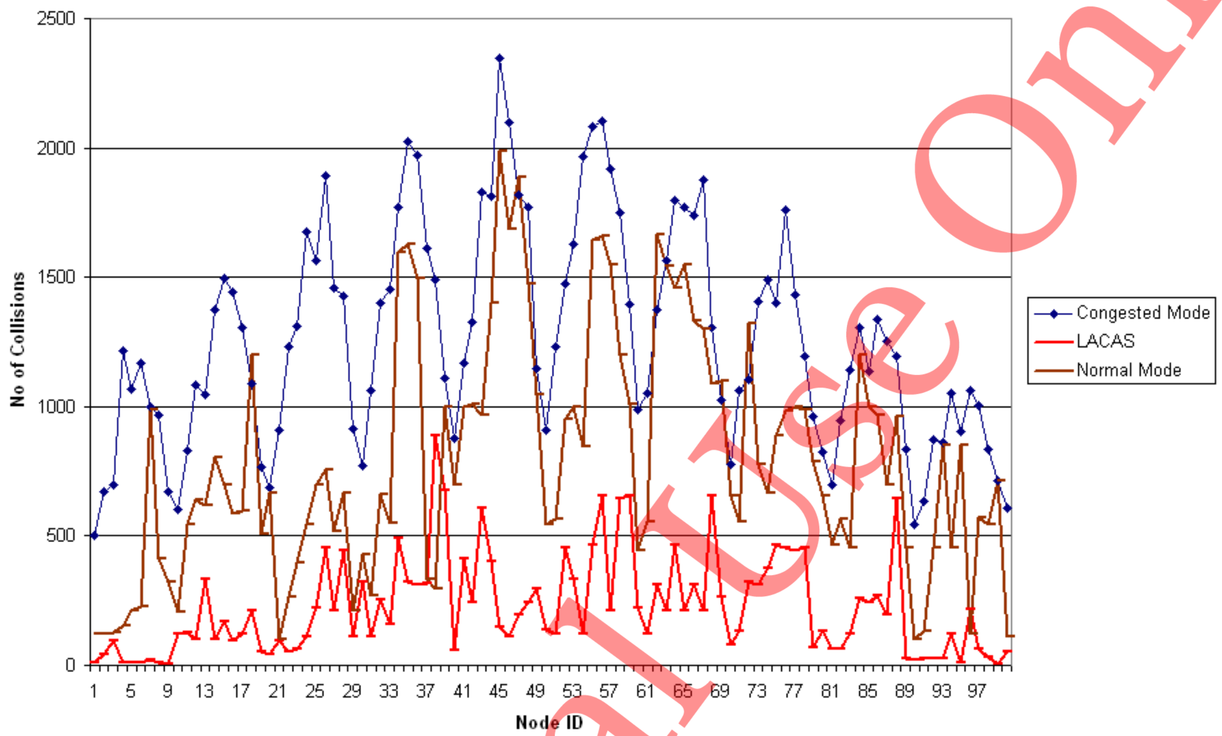

Figure 8: Collisions in the Network

In this case, as well, we find that the levels of collisions are far more pronounced in the normal and the congested modes compared with LACAS. We observe from Figure 7 that the maximum value of the number of collisions in LACAS is just 888, but in the congested and the normal modes it is 2,348 and 1,985, respectively. The significant improvement in reducing the number of collisions in LACAS, testifies to its effectiveness as a congestion avoidance algorithm for use in WSNs.

Throughput: The comparative results in the normal mode, the congested mode and with LACAS are shown in Figure 9. In the congested mode, there is an uncontrolled flow of packets. Therefore, an intermediate node receives a significantly increased number of packets and even transmits accordingly. With LACAS, the rate of flow of packets into and out of a node is controlled by an automaton. Therefore, the number of packets a node receives and the number it transmits is lesser compared to when there is no control of the flow of packets. Increased levels of throughput (i.e., signifying the successful delivery of packets) are desirable. As shown in Figure 9, the ratio between the numbers of packets received at a node with that of the numbers of packets that are transmitted in our study shows that the throughput in case of LACAS is higher with respect to that observed in the normal and the congested modes.

Figure 9: Throughput Performance Results

We observe that the ratio between the received packets and the transmitted packets with LACAS is quite high and is nearly equal to 0.8. In the normal mode, we observe fluctuations in the throughput pattern, but it can be easily inferred that the aggregated ratio is somewhat near to 0.55. In case of the congested mode, this ratio is very poor and it is nearly equal to 0.1. With LACAS, successful packet delivery is quite high because an automaton at a node is selecting the rate that is optimum with respect to the number of packets dropped. This leads to reduced queuing of packers, reduced levels of collision and, in effect, increased throughput.

Number of Packets Delivered to the Nodes / Unicast Packets Received: The total number of packets delivered to the intermediate nodes and the unicast packets received by the nodes are shown in Figures 10 and 11. In unicast delivery, data packets are sent between two designated nodes. While Figure 11 shows only the unicast packets delivered to each of the nodes in the network, the total number of packets (unicast and broadcast) delivered is shown in Figure 9. From a close inspection of Figure 10, it is evident that with LACAS, the number of packets delivered is much larger compared to the execution in the normal and the congested modes.

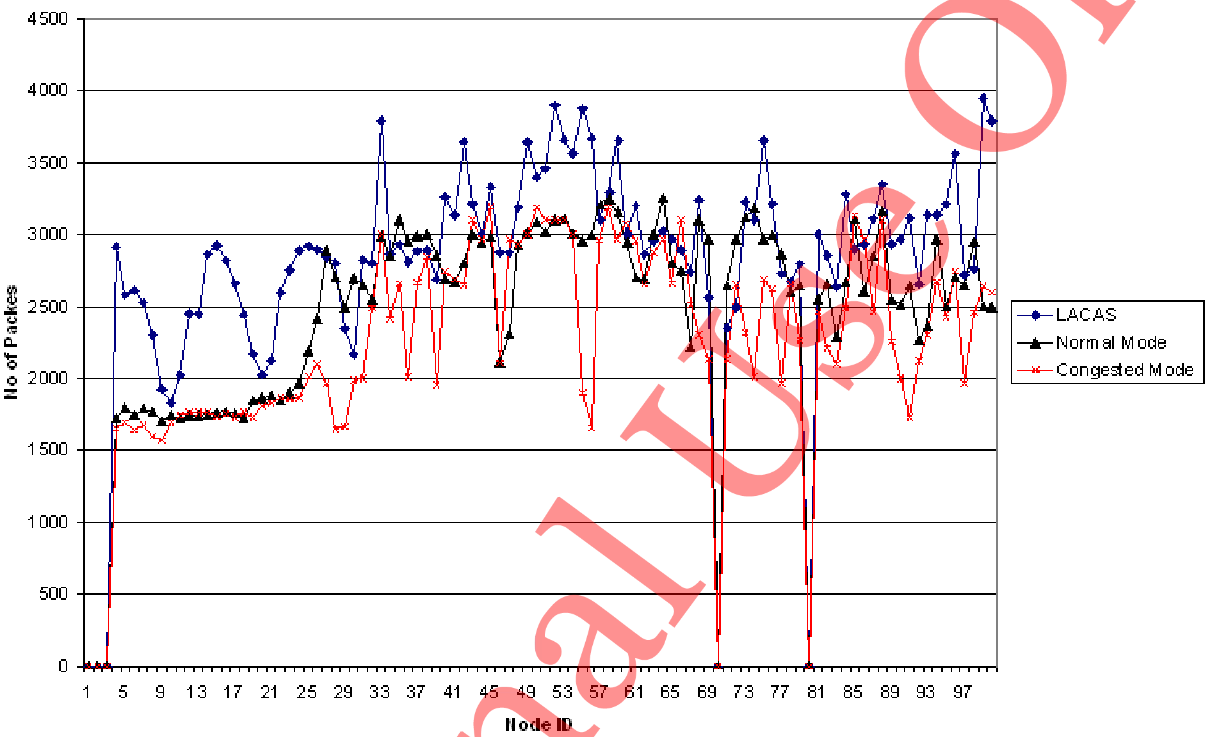

Figure 10: No of Packets Delivered to the Node

The number of unicast packets delivered gives us an indication of the effect on congestion due to unicast packets between two nodes only. When two or more nodes send unicast packets, collisions between different packets are likely to take place and this likelihood increases as more and more nodes get involved. In cases of collisions, not all the packets destined to a particular node reach their target. The more the number of unicast packets received at the nodes, the lesser is the number of collisions in the network, which leads to greater reliability and energy efficiency.

From Figure 11, it can be inferred that, with LACAS, the number of unicast packets received by the sensor nodes is more than that in the other two modes.

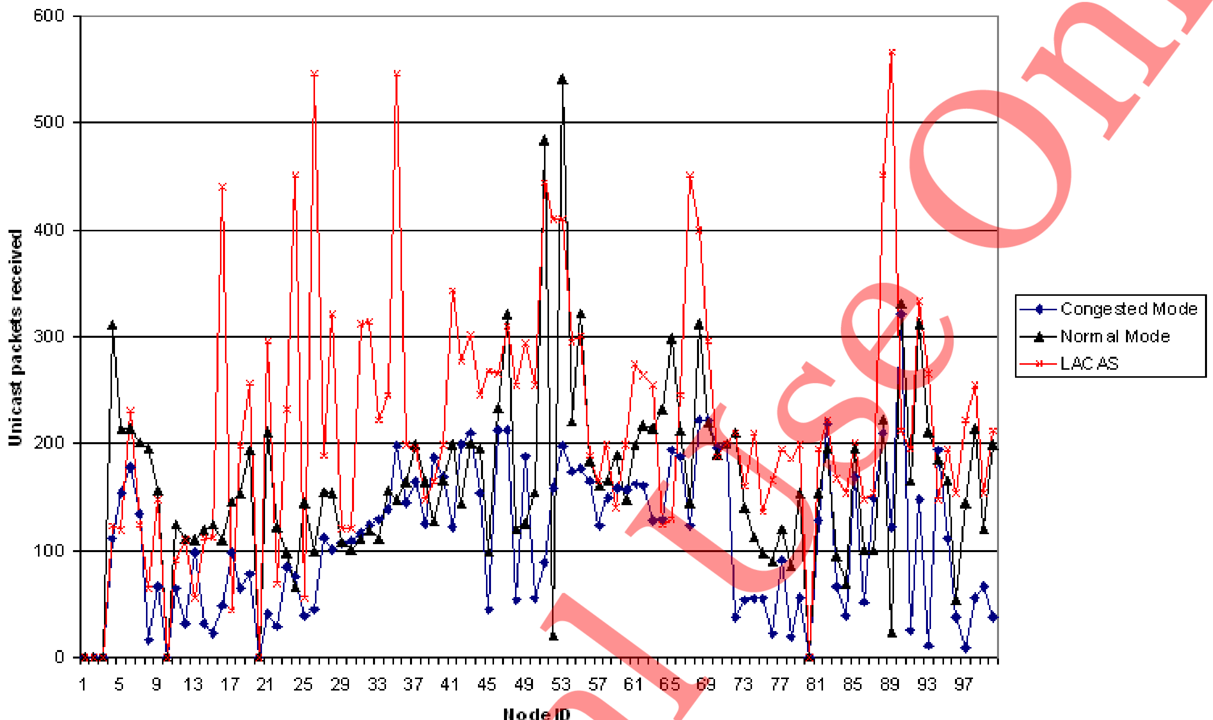

Figure 11: Unicast Packets Received

Variation in the Number of Collisions with Change in Transmission (Tx) Range of the Nodes in LACAS: In the previously presented results, the radio transmission range was set to 10dBm for all the nodes in the network. We were also interested to observe the variation in the collision pattern with the change in the Tx range of the nodes when LACAS is executed. Let us consider 4 nodes, namely, Node A, Node B, Node C and Node D, in which A acts as the source, D as the sink and the remaining ones as the intermediate nodes. Initially, the Tx range was such that the packets transmitted by A will be captured only by B or C but not by D. In this scenario, there exists less traffic which, in turn, will result in less collision. After increasing the Tx range of Node A, it not only gives some packets to Node B as well as to Node C, but also some of the packets are directed towards Node D as well. In this scenario, the number of collisions will increase resulting in the unwanted loss of energy and the reduction in throughput. We collected the results of collisions for 3 different transmission ranges, i.e., 10dBm, 15dBm and 20dBm plotted them as shown in Figure 12. We observed that the pattern of the plots is somewhat similar in all the three cases, but the aggregated number of collisions increase with the increase in the Tx range.

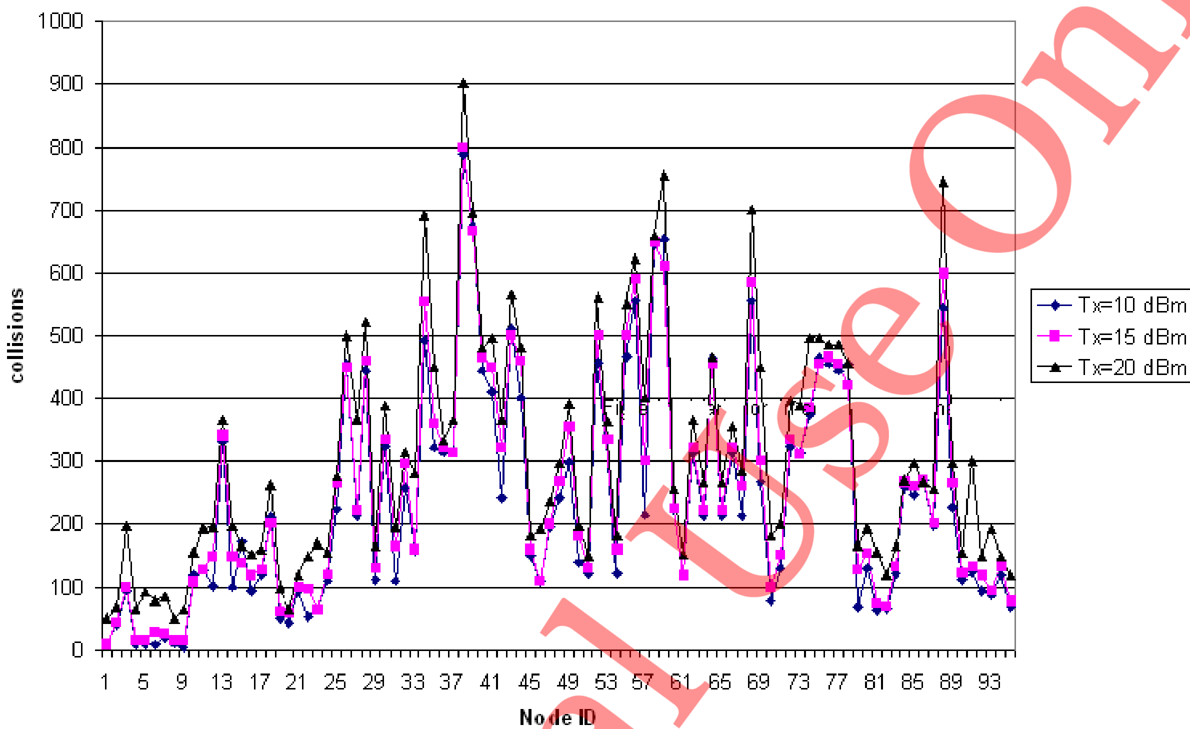

Figure 12: Collisions with Change in Tx range for LACAS

Energy Consumed Versus Tx Range in LACAS: We were also interested to observe the variation in the consumption of energy with the variation in the Tx range when LACAS is executed. The results obtained are shown in Figure 13. As seen in Figure 12, an increase in the Tx range of a node increases the area to which the data packets can be sent, which, in turn, results in an increase in the collision of packets. As the collisions increase, the energy consumed by each node also increases. Increased consumption of energy at the sensor nodes is undesirable, because the nodes are difficult to be recharged in most healthcare applications of WSN. The loss of energy of a node will result in its decreased lifetime of the nodes, which will render them from being used uninterruptedly for longer periods of time, as is mandated in most healthcare applications.

From Figure 13, it can be inferred that when the Tx is 10dBm, the energy consumed by the sensor nodes is 23.5mWh (approx.). When we increased the Tx to 15dBm the energy consumption increases to 25.5mWh (approx.). Finally, when the Tx was increased to 20dBm, it was found that the energy consumption increases to 26.2mWh.

Figure 13: Variation in Energy Consumption with Change in Tx Range in LACAS

Variation in Throughput with the Change in Tx Range: In this Section, we tried to observe the variation in the throughput for LACAS with change in the transmission range of the nodes. As in the case of the previous two experiments, we used the same three Tx ranges, namely, 10dBm, 15dBm and 20dBm. The number of data packets transferred from one node to other and the aggregated throughput reduces with the increase in the Tx range. For 10dBm Tx range, the ratio between the transmitted packets and the received packets varies within the range of 0.6 – 0.9, which shows that most of the packets are successfully transmitted. In the second case, when the Tx range is 15dBm, the above stated ratio decreases to 0.55. Finally, when the Tx range is increased to 20dBm, this ratio changes to 0.4 approximately. These observations can be had from the plot presented in Figure 14.

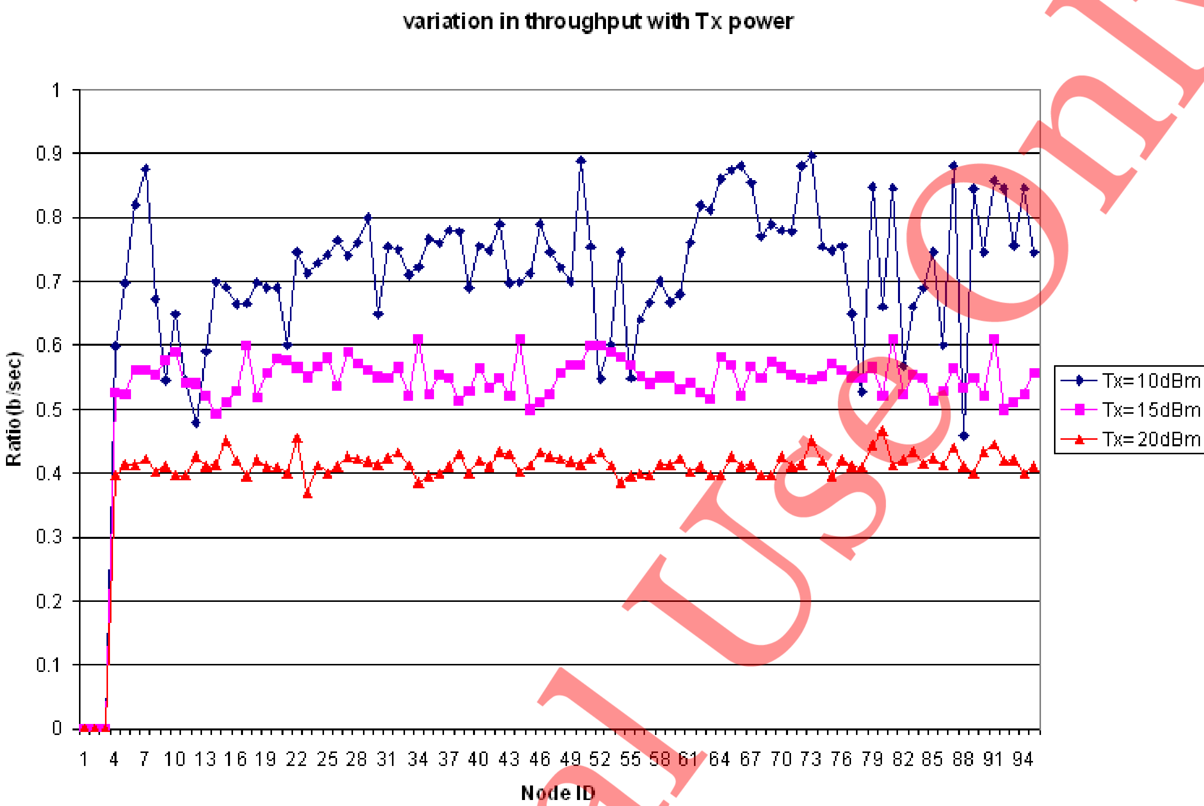

Figure 14: Variation in Throughput with the Change in the Tx range in LACAS

Signals Received and Forwarded to MAC: When a sensor node sends some data packets for another node, the packets traverse a path through the different layers, including the MAC layer, of the protocol stack of the corresponding nodes. Figure 15 shows the number of signals received and forwarded by the MAC layer. The importance of this plot lies in the fact that it gives us an indication as to how much the MAC layer is affected, in terms of the flow of signals through it, by the change in the implementation of LACAS, compared to the normal and the congested modes. We observe that, in LACAS, the number of signals engaging varies between 6,000 and 9,000, compared to 3,000 and 8,000 in the normal mode, and 2,500 and 6,500 in the congested mode. Therefore, we can conclude that, on an average, LACAS does not significantly affect the normal functionalities of the MAC layer.

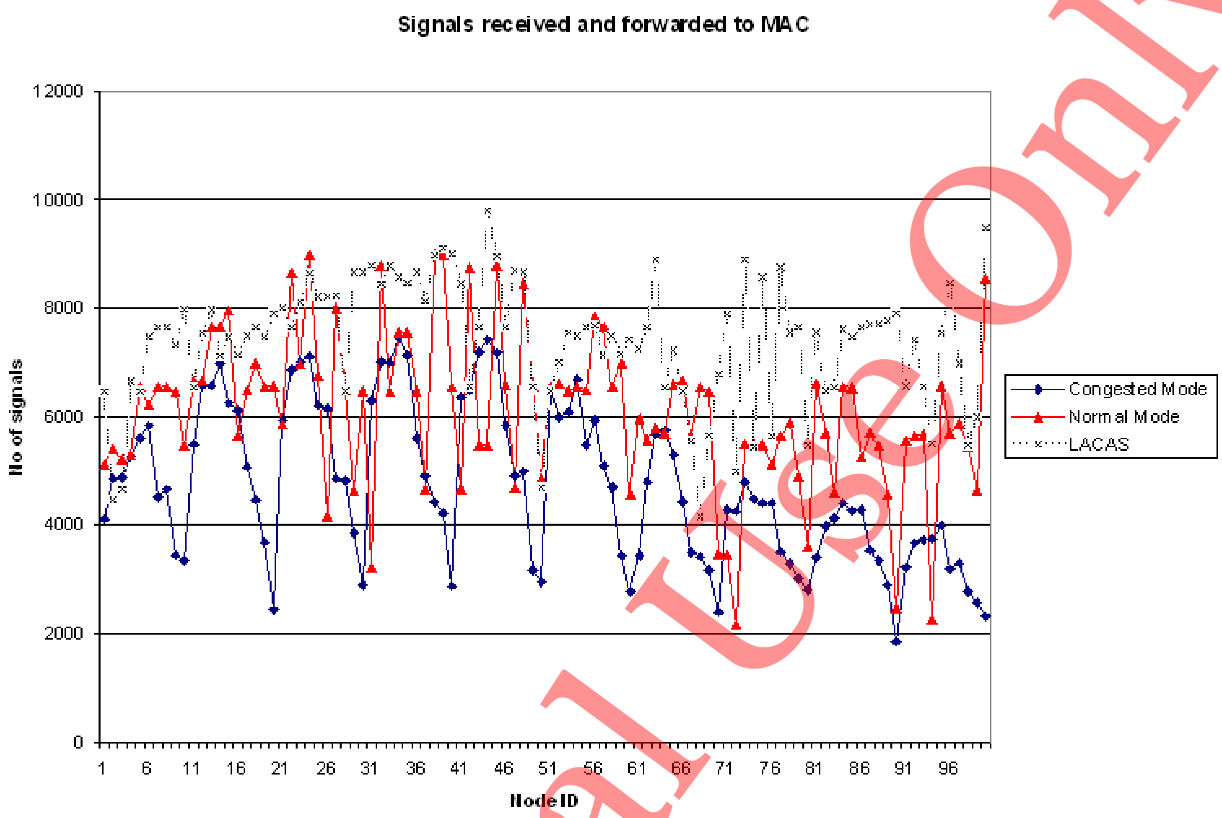

Figure 15: Signals Received and Forwarded to MAC layer

Variation of Throughput in LACAS with the Change in Node Placement Type: The previous experiments were undertaken by keeping the placement of the nodes as uniform. A very high level explanation of uniform, random and grid placement of nodes is provided below. In *uniform placement*, the terrain is divided uniformly into a number of cells, based on the number of nodes in the network. The nodes can be stationed anywhere in those cells. In *random placement*, the nodes are placed randomly anywhere within the physical terrain. Finally, in *grid placement*, the node placement starts from (0, 0). The node number has to be a square of an integer for grid-based pattern. So, 100 nodes in a network were sufficient for the grid placement type. In this experiment, we changed the location of the nodes based on their node placement type and observed the variation in throughput for the nodes in the network, as shown in Figure 16. The results show that throughput depends significantly on the positioning of the sensor nodes.

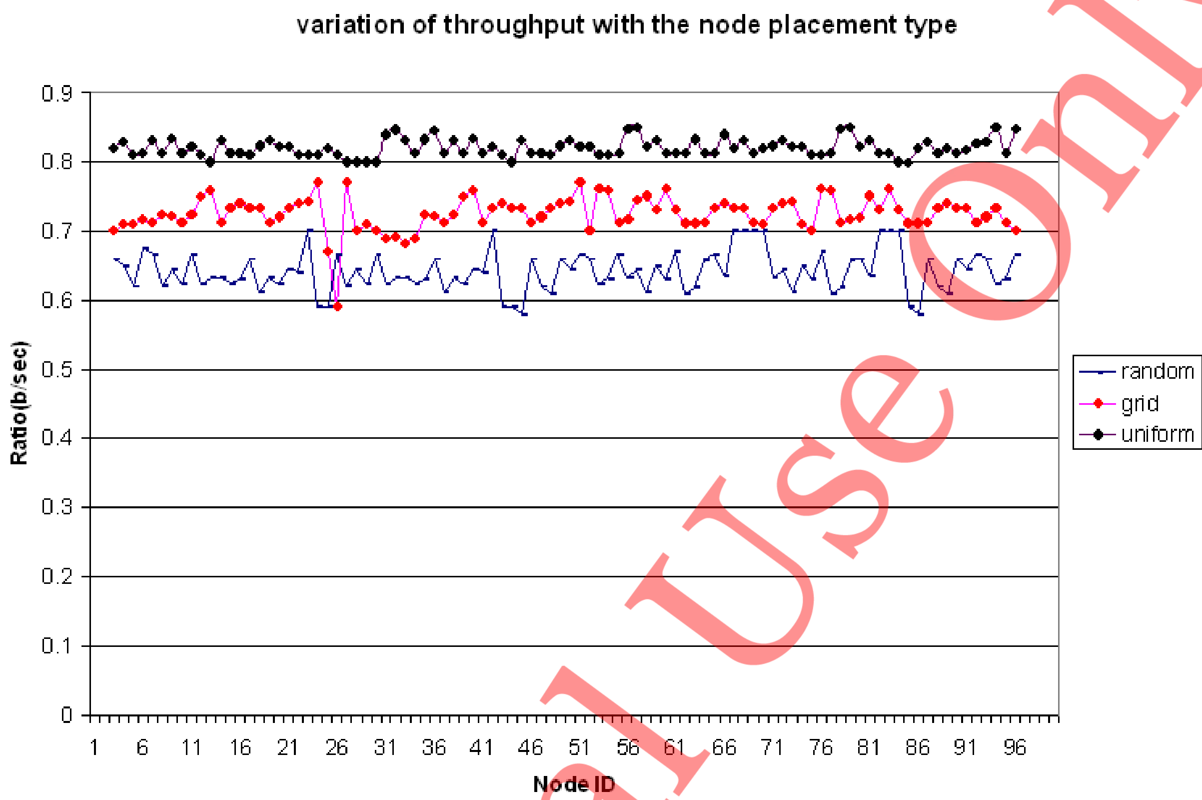

Figure 16: Variation in Throughput with Node Placement Type

For uniform placement type, the ratio was best of the three, close to about 0.8 to the precise. In the grid placement type, the ratio fluctuated in the vicinity of 0.7 and for the random topology it varied between 0.6 and 0.68.

7. Conclusions

To conclude, we have designed a novel congestion control algorithm, suitable for use in healthcare applications. The proposed scheme is capable of effectively avoiding congestion while increasing the throughput, the number of packets delivered to the sink node, while consuming significantly less amount of scarce energy available at the nodes and decreasing the number of collisions at the intermediate nodes. Our approach is underpinned in the principles of Learning Automata (LA). The advantage of our proposed algorithm, LACAS, is that, the automata stationed in the different sensor nodes interact with their environment to select a locally optimal action at every time instant, based on the knowledge of previous levels of congestion, to finally offer a globally optimal solution, capable of reducing congestion significantly, for the future. We conducted extensive simulation experiments to assess how LACAS behaves, with respect to metrics such as the energy consumed, throughput and collisions, for different network settings and operating conditions typical of WSN healthcare applications. As stated earlier, our results show that LACAS is, indeed, capable of curbing down congestion in WSNs. Also, the number of packets that are delivered from one node to other increases significantly.

Let us now present some of our thoughts for future work. For the implementation of our algorithm, we took P-model [25, 28] for the simplicity. In this model, the environment offers only binary responses for any action selected by the automaton. We are currently trying to observe how the Q-model, in which the environment response can take place multiple values in the interval [0, 1], would affect our proposed solution approach. The interest for Q-model is that it is representative of a set of healthcare applications such as monitoring of emergency rescue vehicles such as ambulances. Also, we did our experiment for stationary environments where the penalty probability does not change with time. But, it would be interesting to observe how our approach needs to be changed for non-stationary environments. This particular work is limited to WSNs with stationary nodes. However, we are planning to extend the work for mobile sensors in the future. Finally, we have seen promising simulation results for our proposed algorithm, it would be interesting to assess and compare usefulness and efficiency of our proposed LA-based algorithm in a multitude of real-life current generation and future generation healthcare WSN applications.

References

[1] F. Zhao and L. Guibas *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kauffman, 2004.

[2] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, A. Wood, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges", In *Proceedings of the High Confidence Medical Device Software and Systems (HCMDSS) Workshop*, Philadephia, PA, USA, June 2005.

[3] O. Aziz, B. Lo, R. King, G.-Z. Yang, A. Darzi, "Pervasive Body Sensor Network: An Approach to Monitoring the Post-operative Surgical Patient", In the *IEEE Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks 2006*, pp. 13-18, April 2006

[4] C. R. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, N. Burkhart, A. D. Minassians, G. Dervisoglu, L. Gutnik, M. B. Haick, C. Ho, M. Koplow, J. Mangold, S. Robinson, M. Rosa, M. Schwartz, C. Sims, H. Stoffregen, A. Waterbury, E. S. Leland, T. Pering, P. K. Wright, "Wireless Sensor Networks for Home Health Care", In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops* (AINAW'07).

[5] L. J. Celentano, *RFID-Assisted Wireless Sensor Networks for Cardiac Tele-healthcare*, Master of Science in Computer Engineering Thesis, Rochester Institute of Technology, Rochester, NY, USA, 2007.

[6] B. Lo, L. Atallah, O. Aziz, M. El ElHew, A. Darzi and G.-Z. Yang, "Real-Time Pervasive Monitoring for Postoperative Care", In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks 2007* (*BSN 2007*), pp. 122-127, 2007.

[7] A. Woo and D. C. Culler, "A Transmission Control Scheme for Media Access in Sensor Network," *Proceedings of ACM Mobicom'01*, Rome, Italy, July16-21, 2004.

[8] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proceedings of ACM Sensys'03*, Los Angeles, CA, Nov 5-7, 2003.

[9] C. Wang, K. Sohraby, V. Lawrence, L. Bo, Y. Hu., "Priority –Based Congestion Control in Wireless Sensor Networks" (to appear), *IEEE international Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* –Vol. 1, SUTC'06.

[10] C.T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," *Proceedings of ACM Sensys'04*, Baltimore, MD, Nov3-5, 2004.

[11] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," *Proceedings of ACM Sensys'04*, Baltimore, MD, Nov 3-5, 2004.

[12] P. Levis, N. Patel, D. Culler, S. Shanker, "Trickle: A Self Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," *Proceedings of 1st Symposium on Networked Systems Design and Implementation*, San Francisco, CA, Mar. 29-31, 2004.

[13] C. Y. Wan, S. B. Eisenman, A. T. Campbell, Jon Crowcroft, "Siphon: Overload Traffic Management Using Multi Radio Virtual Sinks in sensor Networks," In *Proceedings of ACM Sensys'05*, Nov. 2-4,2005, San Diego, CA, USA.

[14] S. Misra and B. J. Oommen, "GPSPA: A New Adaptive Algorithm for Maintaining Shortest Path Routing Trees in Stochastic Networks", *International Journal of Communication Systems*, John Wiley & Sons, Vol. 17, No. 10, 2004, pp. 963-984.

[15] S. Misra and B. J. Oommen, "Dynamic Algorithms for the Shortest Path Routing Problem: Learning Automata-Based Solutions", *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, Vol. 35, No. 6, December 2005, pp. 1179-1192.

[16] S. Misra and B. J. Oommen, "An Efficient Dynamic Algorithm for Maintaining All-Pairs Shortest Paths in Stochastic Networks", *IEEE Transactions on Computers*, Vol. 55, No. 6, June 2006, pp. 686-702.

[17] B. J. Oommen, S. Misra and O. -C. Granmo, "Routing Bandwidth Guaranteed Paths in MPLS Traffic Engineering: A Multiple Race Track Learning Approach", *IEEE Transactions on Computers*, Vol. 56, No. 7, July 2007, pp. 959-976.

[18] A. Vasilakos, M. P. Saltourous, A. F .Atlassis , W. Pedrycz, “ Optimizing QoS Routing in Hierarchical ATM Networks Using Computational Intelligence Techniques,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 33, No. 3, August 2003.

[19] J. Praveen, B. Praveen, T. Venkatesh, Y. V. Kiran and C. Siva Ram Murthy, “ A First Step Toward Autonomic Optical Burst Switched Networks,” *IEEE Journal on Selected Areas in Communication*, Vol. 24, No. 12, December 2006.

[20] G. Papadimitriou and D. G. Maritsas, “Learning Automata Based Receiver Conflict Avoidance Algorithms for WDM Broadcast-and-Select Star Networks,” *IEEE/ACM Transactions on Networking*, Vol. 4, No.3, June 1996.

[21] A. Alyatama, “Dynamic Routing and Wavelength Assignment Using Learning Automata Technique,” In *Proceedings of IEEE Communications Society, Globecom* 2004.

[22] G. Papadimitriou and A. S. Pomportsis, “Self-Adaptive TDMA Protocols for WDM Star Networks: A Learning-Automata-Based Approach,” In *Proceedings of IEEE Photonics Technology Letters* ,Vol. 11, No. 10, October 1999.

[23] G. Virone, A. Wood, L. Selavo, Q. Cao, L .Fang, T. Doan, Z. He, R. Stoleru, S. Lin, J. A. Stankovic, “Alarm-Net: An Assisted Living Centered & Testbed-Oriented Information System based on Residential Wireless Sensor Network,” In *Proceedings of Transdisciplinary Conference on Distributed Diagnosis & Home Healthcare*, 2006.

[24] G. –S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell and F. Cuomo, “Funneling-MAC: A Localised, Sink Oriented MAC for boosting fidelity in Sensor Networks,” In *Sensys’06:Proceedings of 4th International Conference on Embedded Networked Sensor Systems*, pp. 293-306, NY, USA, 2006.

[25] K. S. Narendra and M. A. L. Thathachar, *Learning Automata-An Introduction*, Prentice Hall-89.

[26] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, M. Gerla, *GLOMOSIM: A Scalable Network Simulation Environment*, Technical Report, UCLA Computer Science Department, 1999.

[27] W. Stallings, *Wireless Communications & Networks*, 2nd Edn., Prentice Hall, 2007.

[28] M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis,” Learning Automata: Theory, Paradigms and Applications,” *IEEE Transaction on Systems, Man and Cybernetics-Part B*, Vol. 32, No. 6, pp. 706-709, December 2002.

[29] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, “An Efficient Adaptive Bus Arbitration Scheme for Scalable Shared-Medium ATM Switches,” Computer Communications Journal, Elsevier, Vol. 24, No. 9, pp. 790-797, May 2001.

[30] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis,” Learning Automata-Based Bus Arbitration for Shared-Medium ATM Switches,” IEEE Transaction on Systems, Man and Cybernetics-Part B, Vol. 32, No. 6, pp. 815-820, December 2002.

[31] G. I. Papadimitriou, M. S. Obaidat, and A. S. Pomportsis," On the Use of Learning Automata in the Control of Broadcast Network: A Methodology," IEEE Transaction on Systems, Man and Cybernetics-Part B, Vol. 32, No. 6, pp. 781-790, December 2002.

[32] M. S. Obaidat and G. I. Papadimitriou (Eds)," Applied System Simulation: Methodologies and Applications, Kluwer Academic Publishers, Norwell, MA, 2003.

[33] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou and A. S Pomportsis, " Wireless Networks," John Wiley & Sons, Chichester, UK, 2003.

Biographies

Dr. Sudip Misra is *an Assistant Professor* in the *School of Information Technology at the Indian Institute of Technology Kharagpur*, *India*, and is also *an Adjunct* Professor *in the Department of Computer Science* at Ryerson University, *Toronto*, *Canada*. Prior to this he worked in *Cornell University* (USA), *Yale University* (USA), *Nortel Networks* (Canada) and the *Government of Ontario* (*Canada*). He received his Ph.D. degree in Computer Science from Carleton University, in Ottawa, Canada, and the master and bachelor degrees respectively from the University of New Brunswick, Fredericton, Canada, and the Indian Institute of Technology, Kharagpur, India. Dr. Misra has several years of experience working in the academia, government, and the private sectors in research, teaching, consulting, project management, architecture, software design and product engineering roles.

His current research interests include algorithm design and engineering for telecommunication networks, software engineering for telecommunication applications, and computational intelligence and soft computing applications in telecommunications.

Dr. Misra is the author/editor of over *70 scholarly research papers and books*. He has won *five research paper awards* in different conferences. He was also the recipient of several academic awards and fellowships such as the *(Canadian) Governor General's Academic Gold Medal* at Carleton University, the *University Outstanding Graduate Student Award* in the Doctoral level at Carleton University, and the Canadian Government's prestigious *NSERC Post Doctoral Fellowship*. His biography was also selected for inclusion in the 2006-2007 edition of Marquis Who's Who in Science and Engineering, and the 25th Edition of the Marquis Who's Who in the World, California, USA. A mention about him and his work has also appeared in the November 4, 2006 issue of the Ottawa Citizen newspaper.

Dr. Misra is the *Editor-in-Chief* of two journals – the International Journal of Communication Networks and Distributed Systems (IJCNDS) and the International Journal of Information and Coding Theory (IJICoT), U.K. He is an *Associate Editor* of the Telecommunication Systems Journal (Springer

SBM), Security and Communication Networks Journal (Wiley), International Journal of Communication Systems (Wiley), and the EURASIP Journal of Wireless Communications and Networking. He is also an Editor/Editorial Board Member/Editorial Review Board Member of the IET Communications Journal, Computers and Electrical Engineering Journal (Elsevier), International Journal of Internet Protocol Technology, the International Journal of Theoretical and Applied Computer Science, the International Journal of Ad Hoc and Ubiquitous Computing, Journal of Internet Technology, and the Applied Intelligence Journal (Springer). He was invited to chair several international conference/workshop programs and sessions. He has been serving in the program committees of over a dozen international conferences. Dr. Misra was also invited to deliver *keynote lectures* in around a dozen international conferences in USA, Canada, Europe, Asia and Africa.

Mr. Vivek Tiwari is a final year undergraduate student in the Department of Electronics and Communication Engineering at the National Institute of Technology – Durgapur, India. His current areas of research interests include wireless networking, adaptive learning systems, and modeling and simulation of communication systems.

Mohammad S. Obaidat is an internationally well known academic, researcher, and scientist. He received his Ph.D. and M. S. degrees in Computer Engineering with a minor in Computer Science from The Ohio State University, Columbus, Ohio, USA. Dr. Obaidat is currently a full Professor of Computer Science at Monmouth University, NJ, USA. Among his previous positions are Chair of the Department of Computer Science and Director of the Graduate Program at Monmouth University and a faculty member at the City University of New York. He has received extensive research funding. He has authored or co-authored six books and over 400 (400) refereed scholarly journal and conference articles. Dr. Obaidat has served as a consultant for several corporations and organizations worldwide and is editor of many scholarly journals including being the Editor-in-Chief of the International Journal of Communication Systems published by John Wiley. He is also an Editor of IEEE Wireless Communications. In 2002, he was the scientific advisor for the World Bank/UN Workshop on Fostering Digital Inclusion.

Recently, Dr. Obaidat was awarded the distinguished Nokia Research Fellowship and the Distinguished Fulbright Award.

Dr. Obaidat has made pioneering and lasting contributions to the multi-facet fields of computer science and engineering. He has guest edited numerous special issues of scholarly journals such as IEEE Transactions on Systems, Man and Cybernetics, IEEE Wireless Communications, Elsevier Performance Evaluation, SIMULATION: Transactions of SCS, Elsevier Computer Communications Journal, Journal of C & EE, and International Journal of Communication Systems, Security and Communication Networks, International Journal of Computer Networks and Parallel Systems, . Obaidat has served as the steering committee chair, advisory Committee Chair, honorary chair, and program chair of many international conferences. He is the founder of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS and has served as the General Chair of SPECTS since its inception. Obaidat has received a recognition certificate from IEEE. Between 1994-1997, Obaidat has served as distinguished speaker/visitor of IEEE Computer Society. Since 1995 he has been serving as an ACM distinguished Lecturer. He is also and SCS Distinguished Lecturer. Prof. Obaidat is the founder of the SCS Distinguished Lecturer Program (DLP) and its present director.

Between 1996 and 1999, Dr. Obaidat served as an IEEE/ACM program evaluator of the Computing Sciences Accreditation Board/Commission, CSAB/CSAC. Between 1995 and 2002, he has served as a member of the board of directors of the Society for Computer Simulation International. Between 2002 and 2004, He has served as Vice President of Conferences of the Society for Modeling and Simulation International SCS. Between 2004-2006, he has served as Vice President of Membership of SCS. Prof. Obaidat is currently the Senior Vice President of SCS. He has been invited to lecture and give keynote speeches worldwide. His research interests are: wireless communications and networks, modeling and simulation, performance evaluation of computer systems, and telecommunications systems, security of computer and network systems, high performance computing/computers, applied neural networks and pattern recognition, security of e-based systems, and speech processing. During the 2004/2005 academic, he was on sabbatical leave as Fulbright distinguished Professor and Advisor to the President of Philadelphia University and Former Deputy Director General of UNESCO (Dr. Adnan Badran who became in April 2005 the Prime Minster of Jordan. Prof. Obaidat is a **Fellow** of the Society for Modeling and Simulation International SCS, and a **Fellow** of the Institute of Electrical and Electronics Engineers (IEEE). For more info; see:
www.monmouth.edu/mobaidat