Stochastic Power Modeling of Wireless Sensor Networks for Mission Critical Systems

T. S. Pradeepkumar
School of Computer Science and Engineering
VIT University
India
tspembedded@gmail.com

P. Venkata Krishna
Senior Member of IEEE
Department of Computer Science
Sri Padmavathi Mahila Visvavidyalayam
Tirupati, India
dr.krishna@ieee.org

Mohammad S. Obaidat
Fellow of IEEE and Fellow of SCS
Computer Science Department
King Abdullah II School of Information Technology
The University of Jordan, Amman, Jordan
msobaidat@gmail.com; m.s.obaidat@ieee.org

V. Saritha
Department of Computer Science and Engineering
Sree Vidyaniketan Engineering College
Tirupati, India
sarithav3@yahoo.com

***Abstract*—Wireless Sensor nodes consist of communication devices, physical devices (environmental Sensors), processing unit, memory and radio. Optimizing the power consumed by the sensor nodes is always a challenge. The power consumed during communication is high. Therefore, optimizing the power and energy during communication is really necessary. This paper addresses this issue by implementing stochastic power model of wireless sensor nodes to handle any Mission Critical Systems (MCS). Mission Critical Systems are systems that handle tasks and accomplish the real-time deadlines. If a deadline is not met, something catastrophe may occur and the sensor nodes sleep during critical times which will lead to an unstable system. So instead of going to sleep state, the state changes to idle state to handle critical tasks. In this paper, the motes are characterized using Semi Markov Decision Process (SMDP). Various policies were framed for Non-Critical and Mission Critical Systems. Mission Critical Systems uses nodes that meet the deadlines, thereby, optimizing the power and energy used. Our experimental setup improves the energy efficiency of MCS by at least 25%. The model is validated using Crossbow Sensor motes. Also, the model selects the action in the node in order to suggest the best policy for better energy optimization. The SMDP modeling is solved by Dynamic programming using the value iteration function with discounted rewards. Our results have shown that the nodes can go from active to sleep state for non-critical applications and active to semi-sleep state for mission critical application. Our performance results have shown that 25% more power saving is achieved.**

Keywords—Wireless sensor networks; simulation; Semi Markov Decision Process (SMDP); Markov process; dynamic programming

I. Introduction

Wireless sensor networks play a vital role in monitoring various environmental factors like temperature, pressure, radiation, light, sound, pollution level, among others [1]. These devices are powered by a small battery, with limited lifetime, a small processor, memory and a radio. These devices (motes) are prone to failure frequently and are consuming energy during communication, radio processing and memory [2]-[4]. So, modeling these devices for power management is essential. There are various works that models the motes either in the communication, routing, radio and even processors. Some of the works perform the modeling using queuing models, and Markov models for various components that belong to sensor networks [4]-[7]. For example, [2] states that the nodes were characterized based on semi active and full active mode, which is then analyzed with numerical analysis for energy consumption for sensor operations, transmitting energy and the average energy consumption in the node in the active mode [1]-[7]. One of the paper implements a solar energy harvesting model that improvises the SMAC (an access mechanism for wireless sensors). This is done using a queuing model deriving from duty cycle and throughput [12].

The authors in [13] implemented a dynamic low power listening schemes just uses an analytical model to analyze the energy consumption during polling. Hady et al. [14] deals with Low Energy Adaptive Clustering Hierarchy (LEACH) with centralized sleeping protocol that extends the lifetime of a sensor node by going to sleep mode when there is an insignificant data from the clusters. Also in [15], Wang et al. talks about Sensor network with single hop or multi-hops. Single hop seems to be efficient in topology, managing power, and placing nodes according to the design choice. In [16], model sensor networks based on event trigger mechanism that reveals the correlation in energy between the nodes. Terasson et al. [8] describes the work based on modeling component for sensor node that is demanded by the application in which it is deployed.

Wang et al. [9] suggested the reliability of sensor nodes that depends on the sleep and active mode of a node. Alternating between the modes improves the lifetime and reliability of the node. In [10], a dynamic power management scheme, using scheduled switching modes that improve the battery lifetime after a packet transmission, has been devised. Moreover, this paper addresses the battery recovery effect to handle the power management. Some works handle power consumption models that address a particular manufacture of sensor motes [11]. Most of these papers address the modeling for the components of sensor networks that does not handle the type of the network. We model power management problem using Discounted Semi Markov Decision processes (DSMDP) for Mission Critical Systems. We devise an optimal policy that uses the Markov Decision process, which is solved using Dynamic Programming. The latter computes the utilization factor per energy consumption. More research in sensor networks is carried out in [19]-[28]. Section II describes the problem statement and basics of Semi Markov Decision process along with Dynamic Programming. Section III informs the actual proposed idea. Section IV deals with the implementation and result analysis and finally Section V contains the summary and conclusion of the work.

II. Problem Statement and Formulation

A. Motivation

Energy is consumed by sensor nodes when communicating with other sensor nodes by either transmitting or receiving. Radios, processors, sub systems of sensor nodes also consume energy. Hence, energy is a crucial factor that should be optimized for extending the lifetime of the sensor nodes (Motes). This paper models the energy occupation of sensor nodes that were deployed in Mission Critical Systems (MCS). This work is motivated to:

- Improve the lifetime of sensor nodes for a Mission Critical Systems using stochastic modeling.
- Optimize energy during communication, which will be decided based on the states of the node with reward. SMDPs are used to model the sensor nodes by fixing immediate rewards through policies and actions, thereby, identify the power-hungry areas of sensor nodes.

B. Mission Critical Systems

Mission critical systems (MCS) are systems, which when not meeting their deadlines might cause a catastrophic failure. Some examples of MCS are Nuclear Reactor wherein if the radiation is not arrested within a deadline, it can lead to loss of human lives through radiation leakage. The deadlines associated with a MCS are usually hard deadlines. If any task could not meet the deadlines, it can lead to an unstable system. This paper handles the power management for MCS and compares it to that of a Non-Critical System (NCS).

C. Markov Decision Process (MDP)

MDPs consist of set of policies, probability transition matrices, an objective function, reward functions (matrices) and above all a decision maker. These five components are part of Markov decision framework, which is helpful in solving the MDP.

Let $\mu(i)$ refers to the policy which states that the action selected in the i^{th} state for the policy μ. All policies in the system are deterministic. Probability Transition Matrix (PTM) is unique for each policy chosen. For each transition, there will be a reward, which we call as immediate reward and the average reward will be allowed once the entire transition is made and is represented using Probability Rewards Matrix (PRM). Finally, the decision maker is also an agent or a controller.

D. Model Description

Wireless Sensors nodes usually stay in any one of the states like transmit/receive, sleep/idle or active state. Since this work deals with mission critical and non-critical systems, there should be a convincing factor that decides the importance of these systems. Sensor nodes consume more power during transmitting/receiving data and also when they are active. However, they consume minimal power when they are in sleep or idle state. Idle state [16] almost consumes as equal as receiving state as the entire node is waiting for an input. Fig. 1 shows the basic state transition diagram of a given sensor node.

There are totally five states at a given sensor node can stay in based on the system in which it is deployed.

- For mission Critical Systems, the transition mainly occurs in the idle state rather than in the sleep state. As in sleep state, the power occupation is very low; the sensor nodes may tend not to handle a critical task that needs a service. But being in idle state, almost equivalent to the receiving state, will satisfy the job. If a deadline is not met by a task that comes with a critical deadline, not only the system performance is affected, the system may go to an unstable state.
- For Non-Critical systems, the transition occurs in the sleep state after each operation. As the tasks are not critical, even missing a deadline does not affect the performance of the system, it will simple degrade the system over a period of time.

E. Energy Management in Mote

Sensor nodes usually have a radio, processor, sensing unit and a limited memory; all these components need power to sense. So, this model, uses a utilization factor corresponds to each components of the system. Table 1 depicts the Power model in these states.

Table 1 shows various states of a mote and the status of processor, sensing unit and radio. Depending on the type of the system (MCS or NCS), the utilization factor changes in the Transmit/Receive state. The processor will be half active during this state (s4) for a mission critical system and remains in the sleep state for non-critical systems.

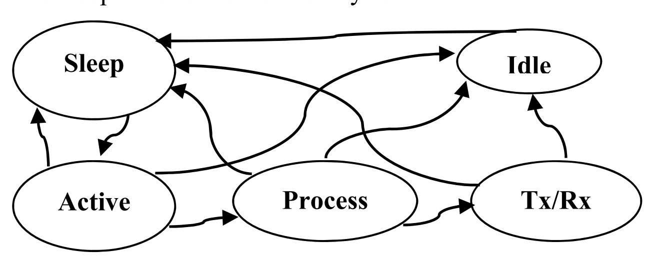

Fig. 1. State transition diagram of a node.

TABLE I. POWER MODE OF VARIOUS STATES

States	P	SU	Radio	UF
Sleep (s0)	Sleep	OFF	OFF	Negligible
Run/ Active (s1)	Active	ON	ON	Xs+Xr+Xp+Xt
Idle (s2)	Half active	ON	OFF	Xs+Xp/2
Process (s3)	Active	ON	OFF	Xs+Xt
Transmit/ Receive (s4)	Half active (MCS) Sleep NCS)	OFF	ON	Xp/2+Xt+Xr Xt+Xr

P – refers the processor status

Su – refers the status of the sensing Unit

UF – refers the utilization factor based on P, SU and Radio.

III. MATHEMATICAL MODEL FOR POLICY EVALUATION

This paper uses the Semi Markov Decision Process (SMDP) modeling to model the states of the sensor nodes. Fig. 1 shows the states of a mote under Mission Critical Systems (MCS) and Non-Critical Systems (NCS).

SMDP consists of: 1) states 2) actions 3) transition probabilities 4) rewards and 5) decision makers. Table 2 shows the characteristics of each components of SMDP.

$$X(i) = R_{iaj} + \int \sum_{j\in|S|}\left[\int_0^y e^{-rt} c_{iaj} P_{iaj}\right] F\, dv \qquad (1)$$

A. Assumptions

The list of assumptions being followed in this paper is

- SMDP says the transition time between the states is not unity, but it is deterministic and non–exponential.
- F is the time distribution in state S (this tells the state at which time the transition should happen).
- Let t_{iaj} represents the transition time between state i to j when there is an action a.
- C_{iaj} is the reward rate.
- Let P_{iaj} be probability of the transition.
- R_{iaj} will be the immediate reward when there is a transition between state i and j.
- R_{iaj}, P_{iaj} and t_{iaj} are stored in the form of a matrix called Time matrix (TM), reward matrix (RM), probability matrix (PM) during transition, respectively.

To solve the SMDP, value iteration algorithm is used instead of policy iteration which needs to be solved for many equations. Value iteration algorithm is easier to compute and can be simulated easily.

The Bellman's optimality equation to solve the value based iteration algorithm using Dynamic programming is given by:

$$T(i) = \max_{a\in A}\left[r_{iaj} - \rho^* t_{iaj} + \sum_{j=1}^{|S|} P_{iaj} \, . T(j)\right] \qquad (2)$$

Where, A(i) is the set of actions.

T* - Set of unknowns just equal to the number of states (S)

TABLE II. CHARACTERISTICS OF SMDP

States	There are 5 states namely sleep (S0), idle (S1), active (S2), process (S3) and Tx/Rx (S4), S={S0,S1,S2,S3,S4}
Actions	In each state, there is an action that transfer from one state to another, a_{ij} means transition from state S_i to S_j when action is a_{ij}
Transition probabilities	The next decision time is determined by the probability distribution P(j)
Rewards	Reward function as mentioned by SMDP is given below as (1)

ρ* - Average reward associated with the optimal (2)

B. Optimal Policy

The policy here is to improve the lifetime of the sensor mote after identifying the power profile of a mote during various states. We use dynamic programming to solve the SMDP for the optimal policy. Two topologies (Star and linear) were tested using Crossbow MTS310 sensor motes.

Model (1) represents the average reward being computed over a period of time based on the number of states. In sensor mote, the reward is allocated based on the following equation:

$$R_{iaj} = \frac{Utilisation\ Factor\ (UF)}{((Power * Transition\ Time) + \in)} \qquad (3)$$

C. Rewards

In this paper, rewards are awarded when a state is transitioned however if the energy is not optimized during a transition then there will be a penalty which is called the discounted reward. Hence, whenever the resource is utilized optimistically, a positive reward is awarded; else there will be a negative reward.

The immediate reward is being given to states during transition. For example, x(S0, a12) represents the immediate reward in state S0 under action a12. From Fig. 1, the rewards are $R_{s1,a12}$, $R_{s2,a21}$, $R_{s2,a23}$, $R_{s3,a31}$, …etc.

The immediate reward for the state diagram of Fig. 1 is shown below. For assumption, the rewards and the discount factor allocated are same for MCS and NCS.

The rewards are normalized on a scale of 0 to 1.0. Since 1 is a perfect system, which is impossible to reach, 0.9 is chosen as the maximum value for active, Tx/Rx state. Also, there will be a penalty if a transition does not happen. For active process and Tx/Rx states, the penalty is high because if a transition does not happen this leads to heavy usage of the resources.

IV. PROBLEM SOLUTION

This paper solves the Bellman's equation using Dynamic Programming approach and hence the model (2) is solved using the Probability Matrix (PM), Transition Matrix (TM) and Rewards Matrix (RM) as given below. For all the states shown in the Fig. 1, the PM, RM and TM are computed and solved using dynamic programming approach.

Rewards are fixed for these cases using Table 3. There will be a discount factor and a penalty also in case the transition does not happen. These cases are solved based only on the states and the actions as specified in the Tuple of SMDP.

TABLE III. REWARD AND DISCOUNT FACTOR

States	Reward(R)	Discount Factor (D)
Sleep (s0)	0 < R < 2	-1 < D < -5
Run/ Active (s1)	R >= 6	-7
Idle (s2)	5 < R < 8	-2 < D < -5
Process (s3)	7 < R < 8	-7
Transmit/ Receive (s4)	7 < R < 12	-7

TABLE IV. SOLVING SMDP

MCS	High Power/availability	H
MCS	Low Power/availability	L
NCS	High Power/availability	H
NCS	Low Power/availability	L

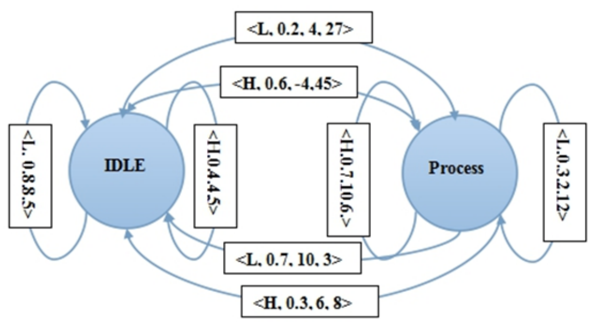

Fig. 2. Idle to process states of MCS.

The MCS and NCS differs by the state idle and sleep, as the MCS have to save power at the same time, the tasks and the deadlines need not be compromised. So MCS rather than going to sleep, it goes to idle mode. Dynamic switching to sleep state is possible at times when there is no demand for a period of time t_D (The time by which the network is not handling any system tasks, if this threshold is reached, the idle state is moved to sleep state).

There are four cases to be solved:

Table 4 shows 4 levels, H, L for MCS and NCS. To solve mathematically, the state idle and process is shown in Fig. 2.

H means high availability and high power whereas L indicates low availability and low power mode. Each of the states process Tx/Rx, and active states go to idle mode whenever the resources need to be used less. So, all the three states move to idle state fewer numbers of times during their process. Fig. 2 represents the state transition diagram between the idle and the process state. The transition is represented by a tuple

$$\langle H/L, Pi, Ri, Ti \rangle \tag{4}$$

This indicates High/Low Power/availability, transition probability, immediate reward and the transition time. Since SMDP is chosen, this has a transition time between the states. The transition is computed for all the states in MCS and NCS and Fig. 2 only show the transition between the idle and process states. The above state is solved based on Bellman's Equation (2) and this is solved using Dynamic programming.

This is an optimality equation, which tells which state is suitable for a given transition and in this work, it tells which transition uses the optimal resource utilization. So, to begin with, let us apply the algorithm of policy evaluation. (Here the policy is to optimize the power and resource usage in a given MCS and NCS System.)

- Let us take the number of iterations be k and the number of states be S.
- Assume that policy is selected arbitrarily and let U be the optimal policy.
- Here in the above equation, there are T^k unknowns and ρ^k unknowns. So, one of the above has to be replaced by 0, since ρ is a reward which is mandatory for the system not have it 0, so the T^k has to be replaced by 0 and we solve for the equation.
- U_k will be the policy selected in iteration k and now a new policy is selected U_{k+1}, such that

$$U_{k+1}(i) = \max_{a \in A} arg\left[r_{iaj} + \sum_{i=1}^{|S|} P_{iaj} T^k(i)\right] \tag{5}$$

- If $U_{k+1} = U_k$, then the selected policy is final and optimal. Else, continue till the optimal policy is reached (where there is no further improvement in the value of iterations).
- Since (2) uses t_{iaj} which is the transition time between the states, it will also be taken in to account during the evaluation.

Fig. 2 shows the states' idle <=> process transition. The other transitions between idle <=> Tx/Rx, and idle <=> Active are also evaluated for policy making. The policy evaluation is done for the non-critical system (NCS) also in order to compare it with the mission critical systems (MCS)

V. NUMERICAL ANALYSIS

The tuple of model (4) shows the results as indicated below. For computation, the values taken are between 0.5to 0.9 in the probability scale, where 0.5 being the lowest and 0.9 being the highest.

For example, in Fig. 2, <L,0.2,4,27> indicates the Low Power mode is preferred, with a transition probability of 0.2 if the idle state wants to move to process state. To maintain low power mode, the idle state tuple <L,0.8,8,5> holds true. So, to maintain, low power mode, being in the idle state will optimize power as the maximum transition probability is 0.8 and the transition probability for moving to process state is 0.2. Hence, the numerical analysis will be based on these factors. Also, the third element in the tuple is the immediate reward which gives the reward based on the power mode, for low power, the reward is 4 for a minimal probability and the reward is 8 for higher probability for the same low power mode. The last element is the transition time (which is special case of SMDP) that does not have a unit time during transition. The transition time is calculated again based on the probability of the switching.

The SMDP model is solved using dynamic programming that shows the results as given below.

The state with probability 1.0 is being tested for unreal case and the limiting probability is negative which shows that such a probability value never exits. Fig. 3 and 4 shows the average reward calculated based on the states with the initial probabilities. In Fig. 3 and 4, in the x axis, there is a value "Actual" which shows the probability as per the actual energy usage by the sensor nodes. For a MCS system, the optimal state for Active is Active; for Tx/Rx, it is SWAP and for process state, it is process state. The optimal state for Tx/Rx is Idle and for idle state, it is Tx/Rx state. For MCS, hence in the Actual mode, the transceiver is used less which may lead to transmit or receive the data that is not preferred in our policy. For NCS system, the Sleep is the optimal state for all other states as all the states go to sleep state after their job is done.

For example, the MTS310 sensor uses power in each state as shown below:

The typical power level of various states of the Mica2 (MTS310) mote is

- Idle State – 270mW
- Sleep State – 10mW
- Active/Run State – 1000mW
- Tx/Rx state – 420mW
- Process State – 620mW

The above values are normalized as actual reward for the system; see Fig. 3 and 4. As per the results, the following Table 5 predicts the actual values. The limiting probability determines the average reward of the given state by being in the same state or transition into other states. In most of the cases, probability with 0.9 works out well.

As per the numerical analysis, the sensors tend to work more if the probability is fixed at 0.9. For this case, assuming the probability of being in a process or active state consumes more power and probability for moving to idle state or sleep state is only 0.1. This makes the sensors work for more amount of time, which might use more power and hence more resources. If the sensors are in the sleep or idle state and do not consume more power, then it may not work and the Mission critical systems may tend to malfunction and such a behavior is not acceptable for these systems. So, the numerical analysis is carried out mainly by considering all of these factors. The actual normalized values are taken for reward calculation.

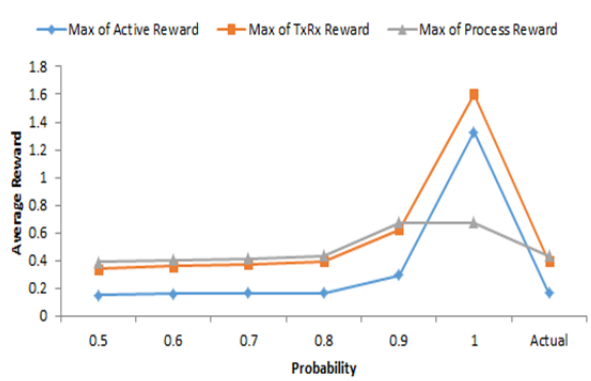

Fig. 3. Average reward for non-critical system.

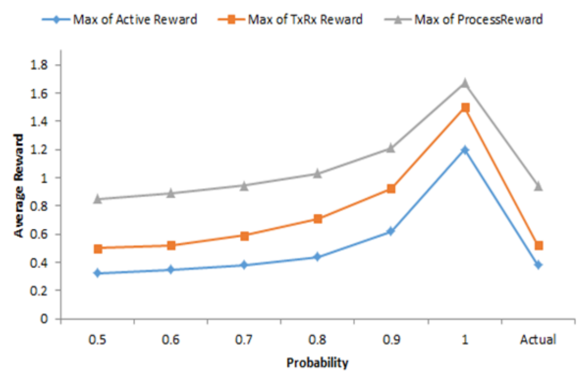

Fig. 4. Average reward for mission critical system.

TABLE V. AVERAGE REWARD FOR 0.9 PROBABILITIES

MCS	***Active***	0.62 for Low Power 0.61 for High Power with P=0.9
	Tx/Rx	0.722 for Low power with p=0.9 0.926 for high power with p=0.9
	Process	1.12 for Low Power with p=0.9 1.1 for high power with p=0.9
NCS	***Active***	0.294 for low power with p=0.9 0.17 for high power with P>0.8
	Tx/Rx	0.62 for low power with p=0.9 0.39 for high power with p=0.9
	Process	0.67 for low power with p=0.9 0.43 for high power with p=0.9

VI. EXPERIMENTAL EVALUATION

We used the MTS310 Sensor motes (Crossbow) for identifying the power profile and power consumption during light duty and heavy-duty sensing areas. Experimentation has been carried out for two kinds of systems, Mission Critical Systems (MCS) and Non-Critical Systems (NCS).

The MCS experiments were carried out in a road to monitor the movement of vehicles in a given time by calculating the vibrations using accelerometer and measuring the sound using mice. The NCS experiments were carried out inside a room to monitor temperature, sound and movement of persons within the room. If no one is there in the room for a period of time, the actuator actuates the power strip that switches off the electrical and electronic devices. The recorded data is reported to the base station through Zigbee IEEE802.15.4, wireless system.

Also for MCS system, Mannasim [16] is used for large number of sensor nodes and the sensing data is collected for temperature and pressure for an aircraft cabin crew. Mannasim is a framework for Network Simulator 2(NS2) [17] used to simulate sensor networks based on temperature and carbon monoxide sensing. The data dissemination, data collection and sensing happens using Gauss theorem. So, the results obtained from Mannasim will be an approximation and results from the real MTS310 will be accurate as the data sensing, and data dissemination occur in real time.

The MTS310 sensor motes has various pins that can be disabled for power saving. In this experimentation, as per Table 1, the components have been enabled/disabled and data is collected and disseminated. In this experiment, the motes form the mesh with two seconds consumes more power and three min mesh formation consumes less power. Star topology is used for mission critical systems where there are 5 nodes, node 1,2,3,4 send packets to node 0 which is a gateway node connected to the internetwork and the packets arrive according to a Poisson distribution.

So, the probability of receiving or transmitting packets after sensing will be calculated by these equations:

$$P_r(n) = \frac{e^{-\rho r T}(T\rho_r)^n}{n!} \tag{6}$$

$$P_t(n) = \frac{e^{-\rho t T}(T\rho_t)^n}{n!} \tag{7}$$

The data is collected based on the arrival rate of the packets during transmission and reception by the motes. The entire process depends on the factors ρ and T.

Motes 1 and 3 are set in low power mode and 2 and 4 are set in high power mode. During high power mode, the packet generation will be higher as more number of packets is generated. Also, the packet generation is directly proportional to the sensing. The motes were sensing the acceleration through accelerometer and vibration through magnetometer sensors. Whenever a vehicle is detected, the accelerometer senses signal and sends it to the nearest node which in turn sends to the gateway node for dissemination. Low power and high-power mode mainly depends on the capability of the mesh, bandwidth and latency, routing and mote radio and topology of the network. The MTS310 sensors were programmed to work as given in Table 1. Control signals were sent to disable sensors and stop the radio. MTS310 does not suspend the processor, so power consumed by the processor as shown in Table 6 is used.

The following results show the energy consumption during state transition from higher power states to lower power states and low power states to high power states, and throughput of packets for MCS and NCS systems.

Fig. 5 shows the energy consumption from active, Tx/Rx, process state to idle in a MCS system. The energy consumption is in the range of 1.2mW to 3.5mW. The active to idle state average is more as the processor, sensor unit and radio are ON in this state (see Table 1). Fig. 6 shows the switching energy for NCS systems when the high-power state transition to sleep state. The average value is more for active state when compared to other states. Also, here Tx/Rx is different from that of the MCS, as in the Tx/Rx state, only the radio unit is on, but in MCS systems the processor consumes half of its power.

The throughput of packet generation for a Mica2 (MTS310) sensor is given below. The sensors generated data for period of 23 hours and it is normalized to a 5s value at steps of 0.1 seconds. The number of packets is generated during each second of information. The average throughput is 9.2kbps. The goodput is also one factor in which the performance of a network can be analyzed. In Fig. 7, the goodness of results happens at a time of 2.6seconds and the packet generated till this interval may not be useful for determining the performance of the network. Also, the goodput is constant beyond a particular period of time.

TABLE VI. PARAMETERS USED FOR TESTING MTS310

Parameter	Value
MAC	IEEE 802.15.4 (Zigbee)
Topology	Star Topology for MCS Linear Topology for NCS
Number of motes	5 motes (1 Base station and 4 Common Nodes)
Period of Simulation	24 hours for MCS (Vehicle vibration and sound) 24 hours for NCS (Monitoring the Room temperature)
Mesh Formation and dissemination interval	Time =2sec (preferred for MCS) & T = 3min (Preferred for NCS)
Power profile	Idle State – 270mW Sleep State – 10mW Active/Run State – 1000mW Tx/Rx state – 420mW Process State- 620mW
Memory	Limited memory footprint

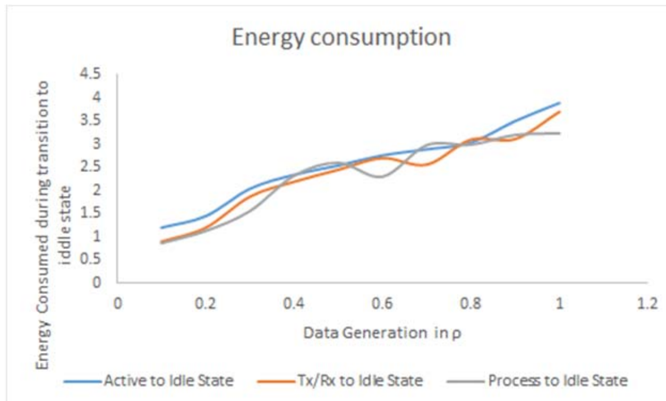

Fig. 5. Energy consumption during transition of states in MCS.

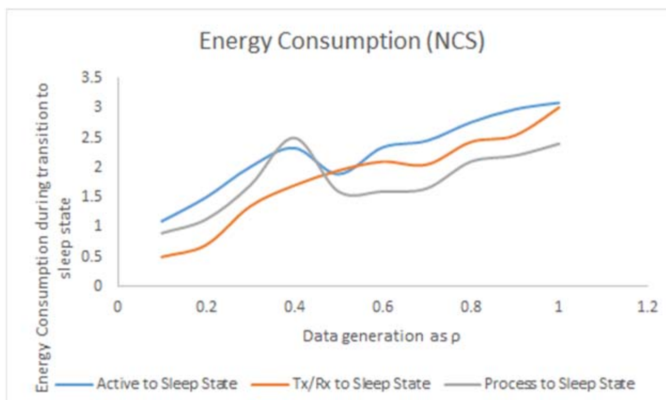

Fig. 6. Energy consumption during transition to sleep state in NCS.

The throughput and goodput is calculated using Mannasim framework [18] for wireless sensor networks by replicating the parameters of MTS310 as given in Table 6.

Some parameters of importance are:

Setting up mica 2(MTS310) mote with Antenna/Omni Antenna and range = 100m

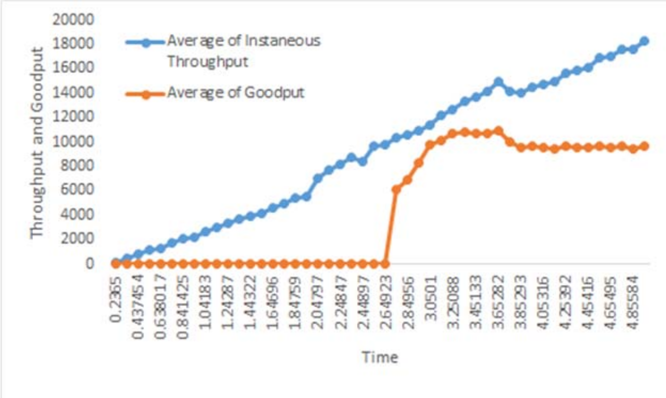

Fig. 7. Instantaneous throughput and goodput of sensor motes.

TABLE VII. PARAMETERS OF NETWORKS USED IN SIMULATION

No of Messages received by the Gateway Node	1820
Average Residual energy of sensor nodes	6.78 joules (initial energy is 7 joules)
Residual energy of Gateway node	8.22 joules (initial Energy 10 joules)
Average delay	3.848 seconds

Nodes count:

- 1 access points
- 4 common nodes
- 0 cluster heads
- 5 nodes
- Phy/Wireless Phy network interface
- Scenario size: 100.0 x 100.0.

The only change is that Mac 802.11 is used instead of Mac 802.15.4 (ZigBee).

Table 7 shows glimpse of the network simulated using Mannasim framework.

VII. CONCLUSION

In this paper, we investigated the problem of energy utilization in Mission critical and non-critical systems using wireless sensor networks. We considered two different topologies: star and linear for MCS and NCS, respectively in order to understand the energy consumption in these topologies. The MCS and NCS were modeled using Semi Markov Decision Process (SMDP) and solved using dynamic programming approach. Immediate Rewards were given based on the utilization factor of the motes, computed average reward using dynamic programming approach and analyzed power consumed areas of both the systems.

An optimal policy that maximizes the long-term usage of motes in MCS and NCS systems before their energy is depleted. Our system performs well when compared to the “Always ON model for MCS or NCS”. We tested the model using MTS310 sensor motes and computed the energy utility for various cases. Under MCS systems, the active and process state need more power to perform the operations needed by the application. Transceiver job is just to transmit and receive the data and enter the idle mode. Under NCS systems, the sleep mode is preferred during non-sensing. However, an NCS system goes to sleep mode rather than idle mode as the system demands that. Also, the motes were simulated in Mannasim Framework that computes the messages received by the gateway node. Throughput and goodness of the packet generation is being analyzed.

Our system performs well when the nodes are minimal in size with a range of 100 x 100 meters. However, our system may not work well in larger scenarios like smart buildings, animal habitat, environmental monitoring, Internet of Things (IoT) Applications, etc. Reinforcement learning is another technique that trains the system by itself and adopts a best policy based on the power history of a network. Also, discount factor can be computed and not taken into our system. This will be extended further as future work.

REFERENCES

[1] M. S. Obaidat and S. Misra,” Principles of Wireless Sensor Networks,” Cambridge University Press, 2014.

[2] Zhang and L, “Modelling and energy consumption evaluation of a stochastic Wireless Sensor Networks” EURASIP Journal on Wireless Communications and Networking 2012:282

[3] Sandra Sendra et al, “Power Saving and energy optimisation techniques for Wireless Sensor Networks”, Journal of Communications, vol 6, No 6, 2011

[4] Carla FAbiana Chiasserini and Michele Garetto, “An analytical model for Wireless Sensor networks with Sleeping nodes”, IEEE Transactions on Mobile Computing, Vol 5, No 12, 2006

[5] Zhi Quan et al, An Efficient Protocol architecture for Large Scale Sensor Networks, IEEE Transactions on Wireless Communications, Vol 6, No 10, 2007

[6] Anders R Kristensen, Dynamic Programming and Markov Decision Processes, Textbook notes of Herd managemenat.

[7] Yuhong Zhang, Wei Li, An Energy based Stochastic model for Wireless Sensor Networks, Wireless Sensor Networks 2011, 3, pp. 322-328

[8] Guillance Terrasson, Renaud Briand et al, “Energy Model for the Design of Ultra Lower Power Nodes for WSN”, Elsevier Precedia Chemistry I (2009), pp. 1195-1198

[9] Chaonan Wang, Liudong Ling et al. “Reliability and lifetime modeling of wireless sensor nodes”, Microelectronics Reliability 54 (2014), pp.160-166

[10] Paulo Sergio Sausen et al. “Dynamic Power management with Scheduled switching modes”, Computer Communications 31 (2008), pp. 3625-3637

[11] A. Prayati et al. “A modeling approach on the TelosB WSN Platform Power consumption”, The Journal of Systems of Software 83 (2010), pp 1355-1363

[12] Navid Tadayon et al. “ Power Management in SMAC based energy harvesting wireless sensor network using queuing analysis”, Journal of Networks and Computer Applications 36 (2013), pp.1008-1017

[13] Sungryoul Lee et al. “Analysis of Dyanmic Low Power Listening Schemes in Wireless Sensor Networks”, IEEE communication Letters, Vol. 13, No. , 2009, pp.43-45

[14] Anar A Hady et al. “Intelligent Sleeping Mechanism for Wireless Sensor Networks”, Egyptian Informatics Journal, 2013, 14, pp.109-115

[15] Qin Wang et al. “A Realistic Power Consumption model for Wireless Sensor Network devices”, IEEE SECON 2006, pp.286-295

[16] Hai Ying, Zhou et al, “Modeling of Node Energy Consumption for Wireless Sensor Networks”, Wireless Sensor Networks, 2011,3, pp.18-23

[17] Web Reference – Network Simulator 2 Software - http://www.isi.edu/nsnam/ns/

[18] Web Reference – Mannasim Framework - http://www.mannasim.dcc.ufmg.br/

[19] Krishna, P. V., & Saritha, V. (2016). Fault-Tolerant Routing in WSNs. Soft Computing Applications in Sensor Networks, 97.

[20] Krishna, P. V., Saritha, V., & Sivanesan, S. (2016). Ubiquitous Context Aware Services. Soft Computing Applications in Sensor Networks, 173.

[21] Arun Nutakki, Venkata Krishna Parimala, Aditya Ahuja, Saritha V, "Finite State Automata Based Fault Tolerant and Energy Efficient Routing (Fsa-Fteer) in Wireless Sensor Networks", The 2011 International Conference on Wireless Networks (ICWN 2011), pp. 81-87, CSREA Press

[22] R. Nanda and P. Venkata Krishna, "Mitigating denial of service attacks in hierarchical wireless sensor networks", Network Security (Elsevier), Volume 2011, Issue 10, October 2011, Pages 14–18

[23] Sudip Misra, P. Venkata Krishna and K. I Abraham, "A simple learning automata-based solution for intrusion detection in wireless sensor networks" Wireless Communications and Mobile Computing, Vol. 11, Issue 3, Pages 426-441, March 2011, DOI: 10.1002/wcm.946

[24] Sudip Misra, Kiran Issac Abraham, Mohhamed S. Obaidat and P. Venkata Krishna, "LAID: A Learning Automata-Based Scheme for Intrusion Detection in Wireless Sensor Networks" Trans. of wiley InterScience journal on Security and Communication Networks, Vol. 2, Issue. 2, pp. 105-115, 2009 [DOI: 10.1002/sec.74]. [SCI level]

[25] R. Nanda and P. Venkata Krishna, "A self enforcing and flexible security protocol for preventing Denial of Service attacks in wireless sensor networks", IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 847 – 850, September 2011

[26] Rohan Nanda, S. Tiwari and P. Venkata Krishna, "Secure and Efficient Key Management Scheme for Wireless Sensor Networks", IEEE Proc. of Computer, Communication and Electrical Technology, Vol.5, 58-61, ICCCET 2011, April 2011, India

[27] Sudip Misra, P. Venkata Krishna, Kiran Issac Abraham, "Energy Efficient Learning Solution for Intrusion Detection in Wireless Sensor Networks", in Proc. of 2010 Second International Conference on Communication Systems and Networks (COMSNETS), pp.1-6, Bangalore, 2010. doi: 10.1109/COMSNETS.2010.5431976

[28] Sudip Misra, Kiran Issac Abraham, Mohhamed S. Obaidat and P. Venkata Krishna, "Intrusion Detection in Wireless Sensor Networks: The S-Model Learning Automata Approach" in Proc. of IEEE International Conference on Wireless & Mobile Computing, Networking & Communication (WiMob 2008) held at Avignon, France, on 12-14 October 2008, pp. 603-607. [DOI 10.1109/WiMob.2008.62]