International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

LIMAC: Long Idle Media Access Control for Wireless Sensor Networks

Adrian Udenze
Department of Electronics and Computer Engineering
Nnamdi Azikiwe University
Awka, Nigeria.

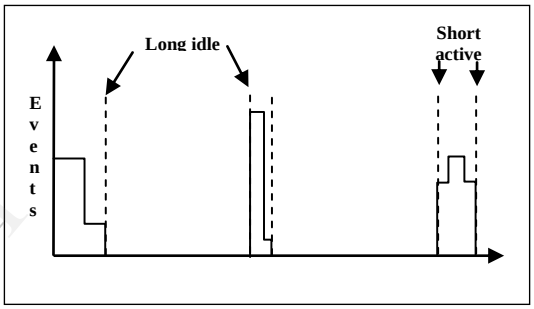
***Abstract*: Event driven traffic patterns in Wireless Sensor Networks are often characterised by bursty behaviour, non-deterministic short busy periods followed by non-deterministic long idle times. This pattern of traffic is more accurately modelled using a combination of probability distributions, an exponential distribution for busy periods and a Pareto distribution with a long tail for long idle periods. The work presented here introduces a novel MAC design named LIMAC. LIMAC has a number of desirable attributes for a MAC including, ability to adapt to changes in network traffic, collision avoidance as well as low computational overheads. In addition, LIMAC takes advantage of long idle periods in WSNs and is suitable for networks with traffic patterns that exhibit bursty behaviour such as in target tracking applications. LIMAC is theoretically based on a Markov Decision Process thus the resulting policies are proven to be optimal. The energy efficiency simulation results for LIMAC under traffic conditions that have long idle periods are shown to outperform HYMAC, a state of the art MAC protocol without any significant degradation in delay.**

Keywords: Wireless Sensor Networks (WSNs); Scheduling; Media Access Control (MAC);

I. INTRODUCTION

Traffic patterns in Wireless Sensor Network (WSN) applications can be thought of as periodic or event driven [1, 2]. For analysis and simulation purposes, periodic data can be modelled using a Constant Bit Rate generator when the bit rate is constant and the Poisson distribution when the bit rate is variable [3, 4, 5]. Traffic patterns for event driven applications such as target detection and target tracking on the other hand exhibit a bursty nature resembling that in figure 1, non-deterministic busy periods followed by long non deterministic idle periods [1, 2, 6]. Such traffic patterns are best modelled using an ON/OFF (Busy/Idle) model where ON Periods are modelled using an exponential distribution and OFF periods modelled using a Pareto distribution with a long tail. Taking advantage of long idle periods by putting nodes into long sleep states can save significant energy in bursty traffic networks and so accurate modelling for simulation and analysis is highly desirable. For example, frame lengths in [7, 4, 8] range between 112ms and 1s. At the beginning of each frame all nodes have to wake up to listen for messages. In a scenario where long idle periods can last up to 5sec, a node that takes advantage of this long idle period by staying in a sleep state saves all the energy needed to wake up and listen for messages for 5 frames which over the life time of a node equates to significant energy savings. If the long idle period lasts for longer which can be the case, then even more substantial energy can be saved.

Figure 1: Bursty traffic



The author's observations are significant for MAC protocol design for the following reasons: 1.) Not all WSN applications are best modelled using an exponential distribution. For some applications, more complex models like the Pareto distribution or a combination of models are required. 2.) Modelling the queue state as in [7] or modelling a stateless system as in [3, 4] is not a suitable representation of the system. Modelling the queue leads to policies that do not take into account the cyclical nature of events, a stateless system design has the same flaw. The work presented here is on LIMAC, a protocol for event driven networks, designed to cope with complex traffic patterns modelled using a combination of probability distributions. LIMAC has a number of desirable qualities for a MAC, like HYMAC [3] and DECMAC [4], LIMAC is able to synchronise and desynchronise active periods so that collisions are avoided. Like HYMAC and RLMAC [7], LIMAC is able to adapt its duty cycle to changes in network traffic conditions. Like HYMAC, RLMAC and DECMAC, LIMAC is based on a reinforcement learning agent so intelligence is at minimal costs. In addition to these qualities, LIMAC is suited to networks where traffic patterns are complex. LIMAC copes with the added complexity by using additional states to account for passage of time as well as a reward mechanism that incentivises nodes to cooperate. LIMAC is based on a Markov Decision Process so policies are proven to be optimal.

The remainder of the paper is organised as follows – a review of related research is presented in section 2. Section 3 describes the LIMAC protocol framework and its associated mechanisms. Section 4 gives simulation results and analysis followed by conclusions in 5.

II. LITERATURE REVIEW

There has been a lot of research into MAC and Scheduling protocol design for WSNs and a review is presented in [3]. Of particular interest to the work presented here are designs based on intelligent agents, in particular Reinforcement Learning (RL) based agents [8]. Their ability to learn optimal schedules with little resource overheads and cope with different traffic patterns and topologies have singled them out as a leading approach to MAC protocol design [3,4,7]. A review of state of the art Quality of Service (QoS) RL based MAC protocols can be found in [5]. In [4] a decentralised RL (DECMAC) scheduling protocol is presented which uses node synchronisation and de-synchronisation to schedule node transmissions such that collisions are avoided. [7] is another RL based protocol in which nodes learn optimal active times within a timeframe during which they send and receive messages i.e. a duty cycle based system in which after an active period, nodes go to sleep for the rest of the timeframe. RLMAC is adaptable to changes in traffic but has no collision avoidance mechanism whereas DECMAC has a collision avoidance mechanism but is not able to adapt to changes in traffic as duty cycles have to be set by a user a priori. Both techniques are combined in HYMAC [3] such that HYMAC is able to adapt to changes in traffic and has a collision avoidance mechanism. All of the afore mentioned protocols however do not take into account the bursty nature of event driven networks and are suited to periodic networks where traffic is modelled using exponential distributions. The works in [1,2,6] show that for event driven networks, exponential distributions do not adequately model network traffic. For target tracking applications for example, the busty nature of traffic requires a combination of probability distributions to more accurately model traffic, an exponential distribution and a Pareto distribution with a long tail.

A. MDP and Reinforcement Learning

Markov decision processes have been used extensively as the theoretical background on which proof of optimality is built for a range of problems in science [9]. WSNs are not an exception and the works in [7, 10] assume a MDP as the underlying system behaviour. A MDP defines a system which can be in any number of states given as a set *S*. In each state *s* a number of decision choices or actions a_s is available. As a result of taking an action a_s in state *s*, a reward $r_{s,a}$ is received and the system transitions to the next state s' with a probability $P_{ss'}$. The goal for a MDP agent is to maximise long term rewards given as

$$V^{\pi}(s) = R(s,\pi,(s)) + \gamma \sum_{s' \in S} P_{ss'}(\pi(s))V^{\pi}(s') \quad (1)$$

Where $\gamma(0 \leq \gamma < 1)$, is a discount factor, $R(s,\pi,(s))$ is the reward function.

The problem defined in equation 1can be solved using several techniques including Value Iteration, Policy Iteration and Linear Programming (LP) where models of the environment, P_{ss} and R exist [9]. RL on the other hand, offers solutions by interacting directly with the environment, observing rewards and subsequently learning optimal behaviour. A comprehensive introduction to RL techniques can be found in [8]. The problem faced by a LIMAC agent is to maximise the rewards it receives $V^{\pi}(s)$ in equation 1 by coming up with an optimal policy $\pi^{*}(s)$. The reward function having been designed to drive agents to take actions that put nodes in the active state long enough to service requests and sleep otherwise. To this end a Q value is maintained for each action choice a_s in each state *s* under a policy π

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P_{ss'}(a)V^{\pi}(s') \quad (2)$$

and updated using Q learning update rule as presented in [8]

$$Q_{k+1}(s,a) = \begin{cases} Q_k(s,a) + \alpha\delta & if\ s_k = s,\ a_k = a \\ Q_k(s,a) & otherwise \end{cases} \quad (3)$$

Where

$$\delta = r_k + \gamma \max_{a' \in A(s')} Q_k(s',a') - Q_k(s,a)$$

III. LIMAC

Figure 2 below shows a LIMAC state diagram model for a node that operates in an event driven wireless sensor network. There are two states ON or OFF. Assume time is divided into frames and subdivided into time slots as in [3,4,7] and media access is duty cycle based. In the ON state traffic arriving at a node is adequately modelled using an exponential distribution. A node in the ON state transitions between an active period where messages are exchanged and a short sleep state. The length of the busy period is governed by a uniform distribution. At the end of the busy period the node enters a long idle period, OFF state, where the next arrival is modelled using a Pareto distribution with a long tail. The tail is derived by using an appropriate filter to filter out short inter-arrival times such that the focus is on the long idle periods. After the first arrival, subsequent arrivals follow an exponential distribution and the cycle repeats. The OFF state is further split into multiple sleep states such that nodes can wake up to check if messages are queued reducing delays. The larger the number of states the more accurate the model is however the more complex it gets requiring more node processor time. The length of time spent in each state is controlled by the LIMAC agent. Thus a LIMAC agent's job is to choose control actions that puts a node into the active state for long enough to service queued requests and to spend idle periods in a low power or sleep state.

For a LIMAC agent to issue the right commands to a node, it has to know what state the environment is in. It was stated earlier that a filter interval is used in the modelling of data to separate short idle periods in the Pareto distribution so that focus is on the long idle periods. This serves as a LIMAC sensor to the real world, thus a LIMAC

agent uses the time between arrivals to check for system state. A LIMAC agent operates as follows.

In the ON state, each time an agent transitions to the short sleep state, the agent decides on a time window in which to be active in the next frame, which translates to choosing a series of time slots. As a result of choosing action a_s, a number of time slots, the agent receives a unit of reward for each slot in which a reception or transmission occurred, otherwise no reward is given. Thus nodes are encouraged to take actions that result in message exchanges. For an exchange to occur two nodes have to cooperate thus, the reward also encourages multi agent cooperation. It is also desirable for nodes to spend all of the active period exchanging messages and to achieve this behaviour, the total reward for each time window, i.e. the sum of the rewards for each slot in the time window, is made a ratio of the total number of reserved slots, equations and 5 below. It is also desirable for nodes to stay awake for long enough to exchange as many messages as possible within a timeframe thus a weighting factor *w* is applied to the sum of rewards within a time frame. Equations 4 and 5 below give the reward function for LIMAC agents in the ON state.

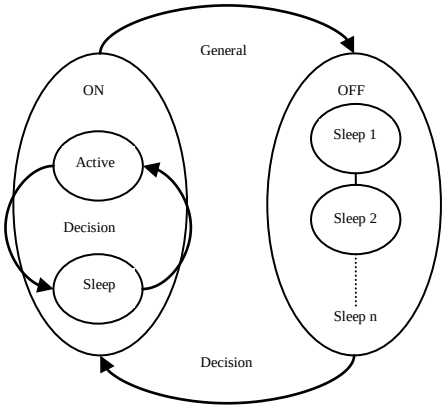
$$\hat{Q} = \sum_{j=0}^{C} Q_{s+j}^{i} \qquad (4)$$

$$Q = \hat{Q}^{w}/_{C} \qquad (5)$$

If a node wakes up at the next designated time and there is no message queued and the time of last arrival is greater than the filter interval, a LIMAC agent assumes a long idle period and transitions to the OFF state where it makes a decision on how long the node should sleep for which again translates to choosing a group of consecutive time slots. As a result of choosing action a_s in state *s* (OFF), the node transitions into the sleep state where it receives a unit of reward for each time slot in which no arrival occurred and no reward for slots in which messages arrived. For the same reasons given for the reward design in the ON state, the total reward is weighted and made a ratio of total sleep time.

The above MDP is guaranteed to converge on optimality given a long enough horizon, proof of which is given in [8], thus a LIMAC agent after an ample exploration phase chooses actions that are optimal for the underlying system. Simulation results presented in the following section show that for systems modelled as described above e.g. for event driven applications, LIMAC outperforms [3] and therefore [4,7] in terms of energy savings with minimal degradation in delay.

Figure 2: LIMAC state diagram model



IV. SIMULATION RESULTS

Simulations were carried out to evaluate the performance of LIMAC compared to HYMAC [3], a state of the art RL based MAC. HYMAC has been shown to outperform DECMAC [4] and RLMAC [7] in terms of energy savings for a given delay constraint in [3]. Models and parameters used for simulations were as follows. Traffic in the ON state was modelled using an exponential distribution with mean inter-arrival times ranging from 1 sec to 0.1 sec. Filters were set to between 1 and 0.1 seconds to separate busy periods from long idle periods corresponding to rate of arrivals in the ON state. Arrivals greater than the filter period were modelled as a Pareto distribution the parameters of which are given in equation 6 below. The *b* value of equation 6 was set appropriately to correspond to the filter size and the α value set to 0.7. The transition from the bursty period to the long idle period was simulated using a uniform distribution with mean ranging from 1to 0.1 seconds. Packet size was set to 50 bytes, power in the active period set to 0.4W, power in the sleep state set to 0.05W [7] and time to transmit a packet was simulated as 20ms.

$$1 - \left(\frac{b}{x}\right)^{\alpha} x \geq b \qquad (6)$$

Two topologies were simulated, a linear topology consisting of 5 nodes and a sink node; and a star topology again consisting of 5 nodes and a sink, figure 3 below. For each topology average delay incurred by packets from generation to delivery at the sink and the average power consumed per second were measured over 50 runs of 5000 seconds simulation time. A sample of the results is presented in figures 4, 5, 6, 7 for simulations with the length of the bursty period determined by a uniform distribution with a mean of 1 second.

Figure 4 shows the average power consumption per second for a node in a linear topology with rate of arrivals in the ON state set as mentioned above, between 1 and 0.5 seconds. Two LIMAC models were simulated, one with two sleep states (LIMAC2) and another with three (LIMAC3). Filter sizes and the Pareto distribution *b* parameter was adjusted appropriately. It can be seen that a

power saving of between 8 (filter size of 0.5 seconds) and 11 % (filter size of 1 second) is achieved for the LIMAC3 protocol compared to HYMAC whereas there is only a 1.3 and 2.5% rise in delay for LIMAC3 compared to HYMAC, figure 5.

Figure 6 shows the average power consumption for the star topology. Two LIMAC agents were simulated as described above, LIMAC2 and LIMAC3. Again there is a 5 to 10 % saving in energy depending on the filter size whereas there is only a 1 and 3% rise in delay, figure 7.

It is worth noting that for longer idle periods in ratio to busy periods, being modelled by adjusting filter sizes, even more power can be saved.

Figure 3: Simulated star and linear topologies

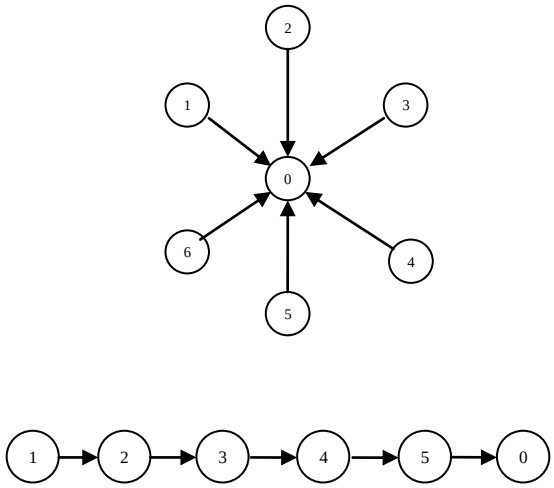


Figure 4: Linear topology average power consumption

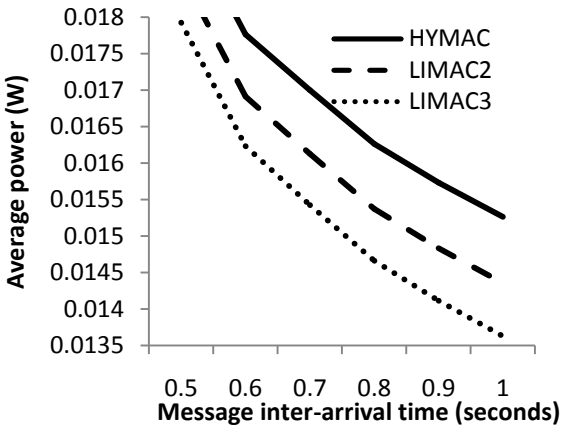


Figure 5: Linear topology average delay

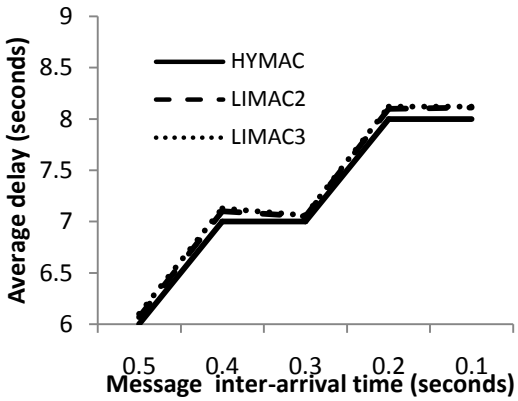


Figure 6: Star topology average power consumption

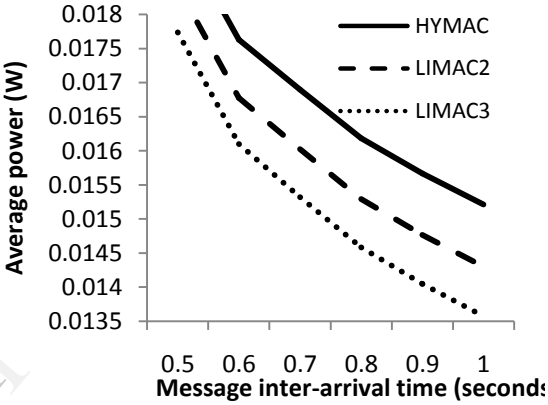
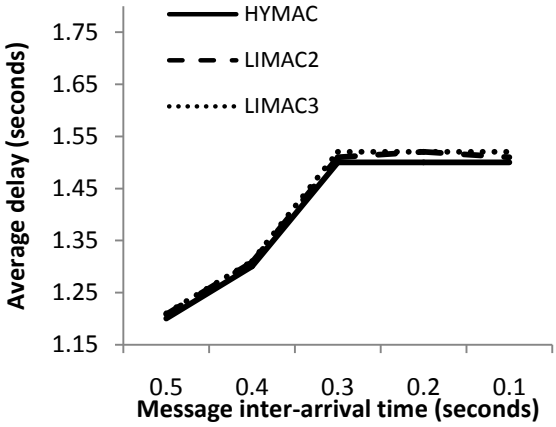


Figure 7: Star topology average delay



V. CONCLUSION

Traffic patterns in WSNs are not always periodic. For event driven networks exhibiting bursty behaviour, complex modelling is required to adequately represent traffic patterns. Under such traffic conditions MAC protocols designed for periodic traffic fail to be optimal. LIMAC is a MAC protocol designed to cope with event driven networks exhibiting bursty behaviour. Simulation results for LIMAC show that in event driven network conditions LIMAC outperforms HYMAC a state of the art MAC protocol designed for periodic traffic, in terms of power consumption. Future work will be aimed at improving the MDP framework by making it event driven.

ACKNOWLEDGEMENT

This research has been supported by Udala R & D Laboratories and Ezned Nigeria Ltd.

REFERENCES

[1] Wang, P. and Akyildiz, I. F. (2009) "Spatial correlation and mobility aware traffic modelling for wireless sensor networks', in Proc. of IEEE Global Communications Conference (GLOBECOM'09.

[2] Wang, Q. and Zhang, T.(2008) "Source traffic modeling in wireless sensor networks for target tracking", in Proc. of the 5th ACM International Symposium on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PEWASUN'08), pp. 96–100.

[3] Udenze, A. (201) "HYMAC: An intelligent collision avoiding MAC for Wireless Sensor Networks", in the International Journal of Engineering

[4] Mihaylov, M., Le Brogne, Y., Tuyls, K., and Nowe, A. (2012), "Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks", International Journal of Communication Networks and Distributed Systems, Vol9, no 3/4, pp 207-224.

[5] Ye, W., Heidemann, J. and Estrin, D. (2004) "Medium access control with coordinated adaptive sleeping for wireless sensor networks", IEEE/ACM Transactions on Networks, Vol. 12, No. 3, pp.493–506.

[6] Simunic, T., Benini, L.,Glynn, P. and Giovanni De Micheli, G. (2001) "Event driven power management" IEEE Trans. On CAD of IC and Sys. 20, pp. 840-857.

[7] Zhenzhen, L. and Itamar, E. (2006), "RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks", International Journal of Sensor Networks, Vol. 1, Issue 3/4 September 2006, pp. 117-124

[8] Sutton, R. S. and Barto, A. G. (1998), "Reinforcement Learning: An Introduction", Cambridge MA: MIT Press.

[9] Puterman, M. L., (1994), "Markov Decision Processes", in Wiley Series on Probability and Mathematical Statistics, J. Wiley and Sons.

[10] Udenze, A. and McDonald-Maier, K. (2011), 'Dyna Routing: Multi Criteria Reinforcement Learning Routing for Wireless Sensor Networks with Lossy Links", in Ad Hoc and Sensor Wireless Networks, pp 285-306.