SMDP-MAC: Event driven Media Access Control for Wireless Sensor Networks

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

ABSTRACT

Duty cycled WSN MACs with fixed periodic wake up and sleep intervals work well for periodic networks. For event driven networks where events can occur at anytime, fixed duty cycles introduce delay waiting for the next wakeup time and power wastage idle listening in the active state. SMDP-MAC, a Semi Markov Decision Process based event driven MAC for WSNs is presented in this work. Simulation results show an improvement in average power consumption and delay for SMDP-MAC compared to LIMAC, a state of the art WSN MAC protocol.

Key Words: WSN MAC, SMDP, RL**.**

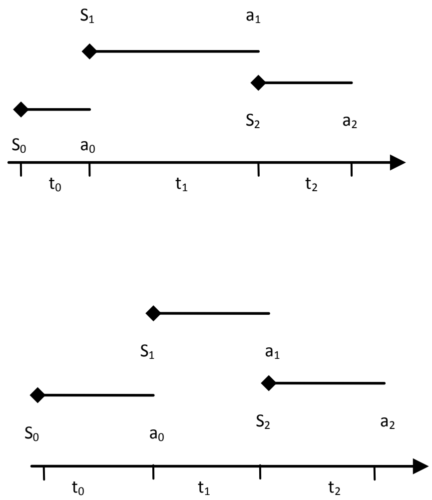
1. Introduction

Periodic networks are networks in which sensor data are transmitted to sink nodes at regular intervals. Nodes wake up at fixed intervals, often within a time frame, to exchange messages and then go to sleep for the remainder of the frame. For such duty cycled networks, a discrete time analysis and modelling of network events works well. The works in Mihaylov et al (2012); Zhenzhen & Itamar, (2006); Udenze[a] (2014) and Udenze[b] (2014), present duty cycled MAC protocols based on a Discrete Time Markov Decision Process (DTMDP). Where models of the network are known a priori, an optimisation problem can be formulated and solved for optimal duty cycles. Where the models of the network are not known a priori, Reinforcement Learning (RL) (Sutton & Barto 1998) have been shown to learn actions online and the learned actions can be proven to be optimal under a DTMDP framework. Event driven networks are networks in which events can occur at any time thus forming a continuous time event driven process. For these networks, the fixed time frame duty cycled model of the discrete time process does not fit well and a new paradigm is need for controlling media access. A Semi Markov Decision Process (SMDP) generalizes the DTMDP by allowing for continuous time event driven analysis. Also, where network traffic patterns are not known a priori, a RL agent can learn optimal policies online. Figure 1 shows a comparison of a discrete time process and a continuous time event driven process. For the discrete time process, decision epochs for each node occur at regular intervals, whereas for the event driven process, decision epochs occur at any point in time.

LIMAC (Udenze[b] 2014) is a RL agent based MAC protocol designed specifically for event driven networks. The work addresses suitable modelling of events in event driven networks and shows that traffic patterns are more accurately modelled when a distinction is made between long idle periods and short busy periods, (Wang & Akyildiz, 2009; Wang & Zhang, 2008), bursty traffic patterns typical of event driven networks. This is in contrast to periodic networks where traffic is accurately modelled by exponential distributions or the Poisson process. Simulation results showed that under bursty network conditions, the LIMAC agent outperforms a HYMAC agent, (Udenze[a], 2014), another

RL based protocol, by having a different duty cycle or policy for long idle periods and short busy ones. The LIMAC decision process is however based on a DTMDP. In the Active state, decisions are made on how long to stay Active and in Long_Idle periods, decisions are made on how long to sleep for. SMDP-MAC is an event driven MAC protocol design that retains the good attributes of LIMAC, collision avoidance and adaptability, and improves on LIMAC by being event driven. In the Active state nodes do not have to make decisions on how long to be active for. Due to SMDP-MAC's event driven design, nodes transition to the active state when there is an event waiting and remain in the active state until the queue is empty before transitioning to an idle state. As a result, nodes spend just the right of time in the active state eliminating the time that would be wasted idle listening. Furthermore, the decision making process is simplified compared to LIMAC and the energy wasted on suboptimal actions during the LIMAC agents exploration stage is removed. SMDP-MAC's analysis is based on a continuous time event driven decision making process, a Semi Markov Decision Process (Putterman, 1994) so that optimality can be proven for resulting policies.

Figure: 1 Comparison of event driven and discrete time processes



2. Literature review

MAC for periodic networks, (Wang & Akyildiz, 2009) are often characterized by duty cycling, (Mihaylov et al 2012; Zhenzhen & Itamar 2006; Ye et al, 2004; Udenze[a] 2014). Nodes wake up at each time frame to exchange messages and then go to sleep for the rest of the frame to save power. This fixed pattern of waking up at regular intervals is well modelled by a DTMDP, (Putterman 1994). Where traffic patterns are known a priori, duty cycles can be set for optimality. Where traffic patterns are not known a priori, duty cycles often have to be learned online using some kind of artificial intelligence, RL, (Sutton & Barto, 1998), has been shown to work well. For proof of optimality of learned policies, a DTMDP is again often assumed. Mihaylov et al 2012; Zhenzhen & Itamar, 2006; Udenze[a], 2014 present results of RL DTMDP MAC protocols for WSNs. Event driven networks, (Wang & Akyildiz, 2009; Wang & Zhang, 2008), are networks in which events can occur at any time, target tracking applications, (Pister, 2000), being an example. The traffic patterns for this type of network is often characterised by bursty behaviour, short non deterministic bust periods followed by long none deterministic idle ones. It is shown in Wang & Akyildiz, (2009) and Wang & Zhang, (2008), that bursty network traffic patterns are best modelled as a combination of independent identical continuous time distributions and not as a Poisson process assumed for a periodic network . In Wang & Akyildiz, (2009) and Udenze[b], (2014), traffic patterns are modelled as a combination of an exponential distribution for the short busy periods and a long tailed Pareto distribution for the long idle periods. The work in Udenze[b] (2014) presents simulation results for a DTMDP solution of the event driven MAC problem. There are two main states ON and OFF within which there are embedded states. In the ON state there are two embedded states Active and Short_Sleep. Nodes in the ON state alternate between Active periods and Short_Sleep periods. In the OFF state there are embedded Long_Sleep states. When a node transitions to a Long_Sleep state, the controlling agent decides how long the node should sleep for. The distinction between the busy ON state and long idle OFF state enables the controlling agent to take a different course of action for each state. This resulted in significant energy savings when compared to HYMAC Udenze[a] (2014), a MAC protocol which assumes a Poisson process. The DTMDP however only acts as an approximation for the underlying event process which is continuous time in nature i.e. an infinite number of states would be required to fully describe the continuous process. SMDP-MAC improves on LIMAC by presenting a continuous time event driven solution to the event driven network MAC problem. SMDP-MAC is based on a SMDP such that the transition between states is modelled by a continuous independent identical distribution and using a suitable learning algorithm, policies can be proven to be optimal. The SMDP is described along with the SMDP-MAC agent in the following sub sections.

2.1 Semi Markov Decision Process

SMDP is a continuous time variant of MDPs and provides a theoretical basis for proof of optimality for systems best modelled using continuous time distributions with or without memory. A distinction is made between states in which decisions can be made and others. At decision epochs, the system is in a state where decisions can be made, at other times the system could be in any number of states. If the system is observed continuously through time, the path observed, commonly referred to as the *natural process,* charts the progression between all states. On the other hand, the *embedded Semi-Markov Decision Process* charts the system progression through decision epochs or states where decisions can be made. To calculate costs and rewards, information is required through all state transitions (natural process), however only the state of the system at a decision epoch is required to make a decision. Let *S* denote a countable state space. At a given decision epoch, the system occupies state $s \in S$ and has to choose an action $a \in A_s$. As a result of choosing action $a \in A_s$ the next decision epoch occurs at or before time *t* and the system state at that decision epoch equals *j* with probability $Q(t, j \mid s, a)$

$$Q(t, j \mid s,a) = P(j \mid s,a)F(t \mid s,a) \quad (1)$$

where $F(t \mid s,a)$ is the probability that the next decision epoch occurs within *t* time units of the current decision epoch, given that the decision maker chooses action $a \in A_s$ in state *s* at the current decision epoch. $P(j \mid s,a)$ denotes the probability that the embedded Markov process occupies state *j* at the subsequent decision epoch when action *a* is chosen in state *s* at the current decision epoch. $p(j \mid t,s,a)$ on the other hand represents the probability that the natural process occupies state *j, t* time units after a decision epoch, given that action *a* was chosen in state *s* at the current decision epoch and the next decision epoch has not occurred prior to time *t*. $F(t \mid s,a)$ may or may not be independent of *s* or *a*. An infinite number of decisions in finite time is avoided by imposing the following assumption:

Let $\varepsilon > 0$ and $\delta > 0$ exist such that $F(\delta \mid s,a) \le 1-\varepsilon$ for all $s \in S$ and $a \in A_s$.
When action $a \in A_s$ is chosen at a given decision epoch, and the system occupies $s \in S$, a lump sum reward (or cost) $k(s,a)$ is received (or paid). Also, a reward (or cost) is received at a rate $c(j',s,a)$ for the duration of the natural process in state *j'* given that action *a* was chosen in state *s* at the preceding decision epoch.
The total expected reward between the two decision epochs for the infinite horizon model is given as:

$$r(s,a) = k(s,a) + \int_0^\infty \sum_{j \in S} \left[\int_0^u c(j,s,a) p(j \mid t,s,a) dt \right] F(du \mid s,a) \quad (2)$$

The total expected reward generated by the process up to time *t*, given that the system is in state *s* at time 0 is

$$v_t^\pi(s) = E_s^\pi \left\{ \int_0^t c(s,a,u)du + \sum_{v=v_0^\pi}^{v_{t-1}^\pi} k_v(s,a) \right\} \quad (3)$$

And the average expected cost for all *s:*

$$g^\pi(s) = \liminf_{t \to \infty} \frac{v_t^\pi(s)}{t} \quad (4)$$

Let *m(j|s,a)* be the probability that the SMDP occupies state *j* at the next decision epoch given that action *a* was chosen in state *s* by the decision maker, then
m(j|s,a) = P(j|s,a) (5)
where *P(j|s,a)* denotes the transition probabilities for the embedded MDP. Therefore for each $a \in A_s$, *m(j|s,a)*is a transition probability function:

$$m(j \mid s,a) = \int_0^\infty p(j \mid t,s,a) F(dt \mid s,a) \quad (6)$$

and $$y(s,a) \equiv E_s^a\{\tau_1\} = \int_0^\infty t \sum_{j \in S} Q(dt, j \mid s,a) \quad (7)$$,

wherey*(s,a)* is the expected time until the next decision epoch given that action *a* is chosen in state *s* at the current decision epoch.
Note that for discounted models, assuming a continuous time discounting rate α>0, where the present value of one unit received *t* time units later equals $e^{-\alpha t}$. Equation (2) above is thus replaced by:

$$r(s,a) = k(s,a) + \int_0^\infty \sum_{j \in S} \left[\int_0^u e^{-\alpha t} c(j,s,a) p(j \mid t,s,a) dt \right] F(du \mid s,a) \quad (8)$$

Theorem (1): The optimal scheduling policy can be found by solving:

$$h(s) = \min_{a \in A} \left\{ \cos t(s,a) - g(s) y(s,a) + \sum_{j \in S} m(j \mid s,a) h(j) \right\} \quad (9)$$

Where *h(s)* is referred to as the bias, *g(s)* is the average cost and *m(j|s,a)* is as defined above. Proof of the theorem and an in-depth study of SMDPs can be found in Puterman (1994). The SMDP described above and its subsequent solution will yield near optimal policies for a system with at most one underlying process exhibiting memory, i.e. the service times or request arrivals.

2.2 Reinforcement Learning

The problem defined in (9) can be solved using several techniques including Value Iteration, Policy Iteration and Linear Programming (LP) where models of the environment, $Q(t, j \mid s,a)$, *m(j|s,a)* equations 1 and 5, are available. Reinforcement Learning (RL) on the other hand offers solutions by interacting directly with the environment, observing rewards and subsequently learning optimal behaviour. Sutton & Barto (1998) offer a comprehensive introduction to RL techniques. The problem faced by a SMDP-MAC agent is to maximise the rewards it receives $v_t^{\pi}(s)$ equation 3, rewards that are 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* in each state *s* and updated using Q learning update rule as presented in Sutton and Barto (1998)

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

Where $\delta = r_k + \gamma \max_{a' \in A(s')} Q_k(s', a') - Q_k(s,a)$ (10)

RL agents make decisions based on taking actions in the environment and reinforcing positive outcomes. Thus the design of a RL based agent involves to a large extent determining adequate rewards to incentivise the agents to take the right actions. The SMDP-MAC RL agent reward is presented in section 3.1.

3. SMDPMAC controller

Assume transmitter events at an arbitrary sensor node by which is meant received and transmitted packets is suitably modelled by a general independent identical distribution. The time taken to transmit and receive the packets is suitably modelled by an exponential distribution. The queue only advances when the transmitter is idle. Thus the queue state of the sensor node follows a M/G/1 queuing system. Assume that the network is synchronised periodically during which time new nodes establish communication with neighbours. Assume that a node's state transition diagram is as presented in LIMAC where there are two main states ON and OFF and embedded states within each main state. See figure 2 below. In the ON state, nodes remain in the Active state exchanging messages until the queue is empty after which it transitions to the short sleep state if the time since the last event is greater than a time out period (T_{out}) but less than a long idle filter period, otherwise the node transitions to the OFF state. The filter period is used to differentiate between the ON and OFF states. In the short sleep state the node decides how long to sleep for and sets a timer, the node transitions to the active state when the time expires. In the OFF state the MAC controlling agent decides how long the node should sleep for and sets a timer. The node wakes up and listens for messages in short sensing periods and will either transition to the ON state if an event is detected or into another sleep state if none is detected after T_{out}. For each action taken the SMDP-MAC gets a reward which is described as follows.

3.1 SMDPMAC reward

An agent's reward for choosing an action *a* a sleep time *t*, in a sleep state *s*; is the sum total of rewards for the duration *t* of a sleep period *s*. *t* is an appropriate function of the node event arrival rate. A reward of 0 is awarded for every arrival in the sleep period and a 1 for every other time step within *t*, equation 11 below. Furthermore, the total reward for the sleep period is weighted, *w* in equation 12, so as to control the desirability of long sleep periods, longer delays lower power, over short ones, shorter delays, higher power.

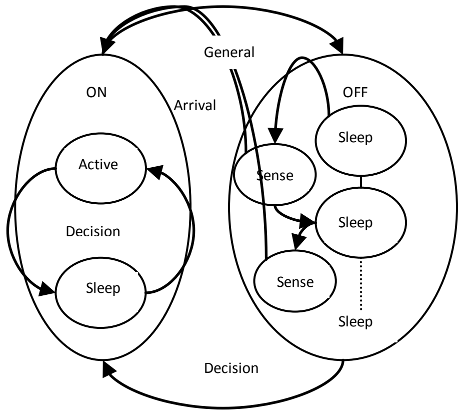
$$\hat{Q} = \sum_{j=0}^{t} Q_{s+j}^{t} \quad (11)$$

$$Q = \hat{Q}^w / t \quad (12)$$

4. Simulation results

Simulations were carried out to compare the energy saving performance of SMDP-MAC and LIMAC, a state of the art RL based MAC protocol, under bursty traffic conditions. Traffic events were modelled as presented in Udenze (2014).

Figure 3: SMDP-MAC controller state diagram



The system state transition diagram is given in figure 2 above. In the ON state, traffic is modelled as an exponential distribution. The transition from the ON state to the idle state is governed by a normal distribution and in the OFF state, events are modelled as a Pareto distribution with a long tail. Two topologies were simulated, a linear topology consisting of 5 nodes and a star topology consisting of 5 nodes and a sink node, Zhenzhen, & Itamar, (2006); Udenze (2014). The rate of traffic arrivals in the ON state was varied between 1 and 10 per second, an inter-arrival time of between 1 and 0.1s. At the same time, the parameters of the Pareto distribution were adjusted appropriately, the filter size was set to between 1 an 0.1s to separate the ON from the OFF period, the α value set to 0.7 and the *b* value adjusted to match the filter size, equation 13 below. The mean of the normal distribution used for controlling the transition between the ON and OFF states was also varied between 1 and 0.1s appropriately. Power in the Active state was set to 40mW and in the sleep state to 30uW, Memsic.com (2015). The size of packets was set to 50 bytes and the time to transmit a packet set at 20ms. The learning rate α equation 10, of the RL algorithm was set to 0.1

$$1 - \left(\frac{b}{x}\right)^{\alpha} \; x \geq b \quad (13)$$

Figure 3 shows the average power consumption for SMDP-MAC and LIMAC averaged over 50 runs of 5000 seconds simulation time. There is a 7% reduction in average power consumption on average for SMDP-MAC compared to LIMAC. This reduction in energy the author puts down to a reduction in the time spent in the Active state due to SMDP-MACs event driven process and also, the simplified decision making process reduces energy wasted on sub-optimal actions during the LIMAC agent's exploration phase. Figure 4 shows the average delay incurred by both SMDP-MAC and LIMAC. Compared to LIMAC, SMDP-MAC reduces delay by 3%. This can again be explained by the event driven nature of the SMDP-MAC protocol and again the simplified decision process. Next a star topology consisting of 5 nodes and a sink was simulated using the same parameters as for the linear topology. Again compared to LIMAC, there is on average a 5% drop in power consumption, figure 5. The explanation for this as with the linear topology is the simplified decision making process as well as the event driven nature of SMDP-MAC. Nodes wake up at irregular intervals reducing the chances of collisions, retransmissions as well as overhearing. In figure 6, delay results are presented, there is a slight drop in delay for SMDP-MAC however at higher data rates the delay results are similar to LIMAC, with a marginal 2% drop.

Figure 3: Average power consumption, linear topology

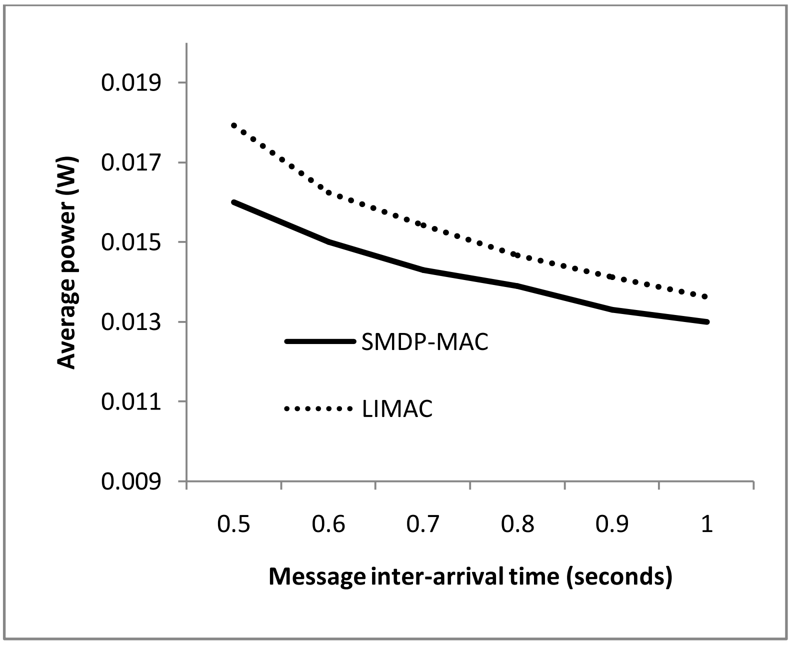


Figure 4: Average delay, linear topology

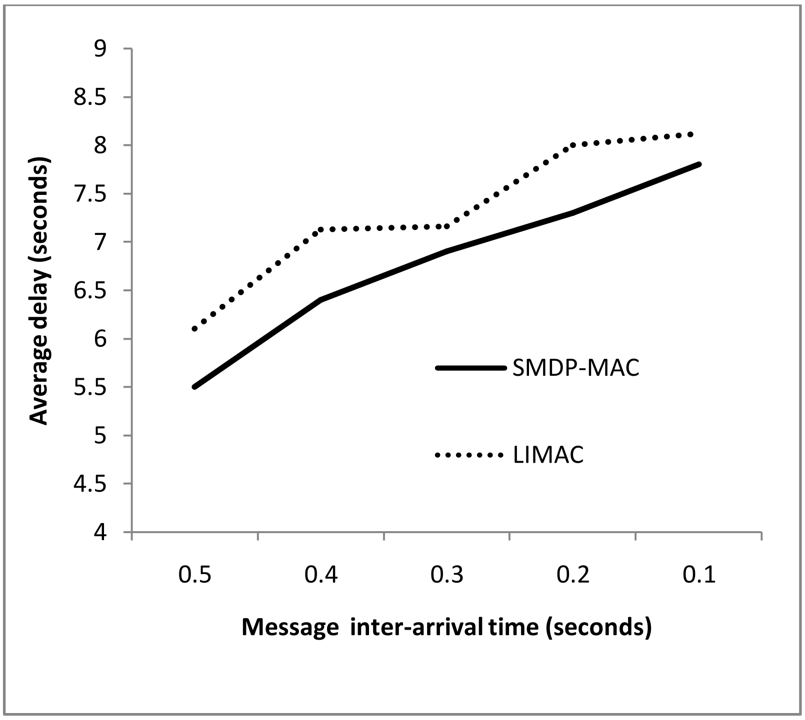


Figure 5: Average power, star topology

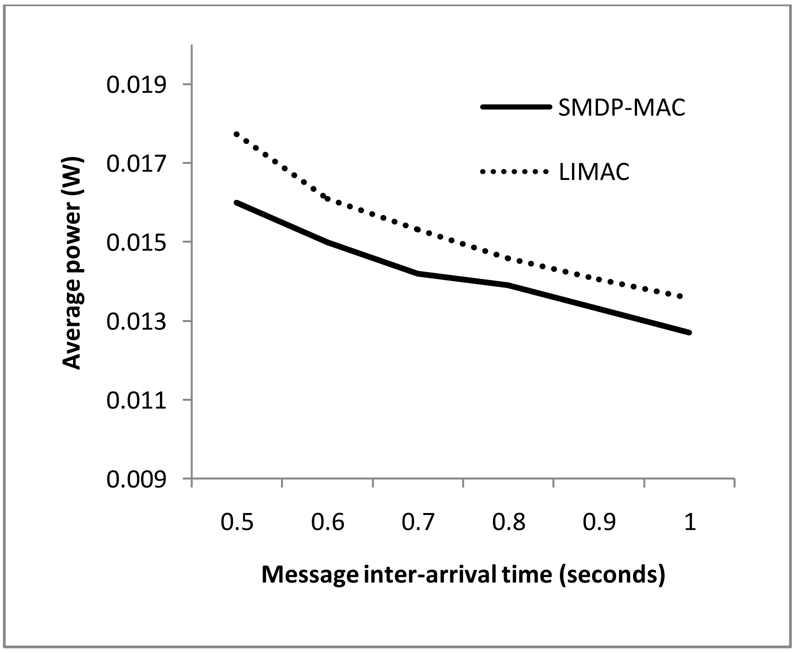
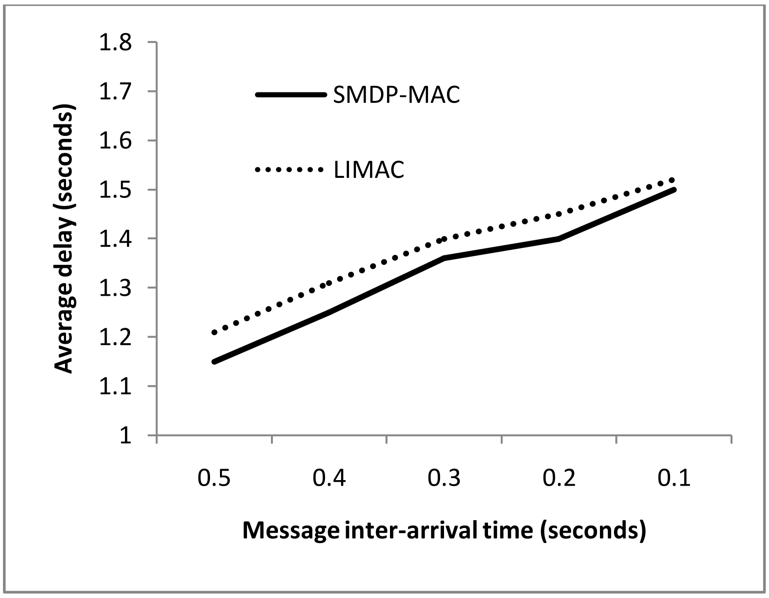


Figure 6: Average delay, star topology



Conclusions

Duty cycled MACs with periodic sleep and wake up times do not accurately reflect the bursty nature of event driven networks. Furthermore, the discrete time analysis often used for modelling and analysing system behaviour in periodic networks does not map well to event driven networks. SMDP-MAC, a RL SMDP based event driven MAC has been shown to outperform LIMAC, a state of the art MAC in terms of power efficiency and delay under bursty network conditions. Future work will involve the study of SMDP-MAC under dynamic traffic conditions.

Acknowledgements

The author acknowledges the contributions of Udala laboratories.

References

[1] 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, Vol 9, no 3/4, pp 207-224.

[2] Pister, K. S. (2000), "Military applications of sensor networks," in *of Institute for Defense Analyses Paper P-3531, Defense Science Study Group*, 2000.

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

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

[5] Udenze, A. (2014), “HYMAC: An intelligent collision avoiding MAC for Wireless Sensor Networks”, in the International Journal of Engineering Research & Technology, Vol 3, Issue 11.

[6] Udenze, A. (2014), “LIMAC: Long Idle Media Access Control for Wireless Sensor Networks”, in the International Journal of Engineering Research & Technology, Vol 3, Issue 11.

[7] 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.

[8] 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.

[9] www.memesic.com/userfiles/Datasheets/WSN/micaz

[10] 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.

[11] 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