RMAC: Renewal theory Media Access Control for periodic Wireless Sensor Networks

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

ABSTRACT

Duty cycling has proven to be an effective Media Access Control strategy for Wireless Sensor Networks. Predominantly for duty cycled networks, frame length determines the frequency at which nodes wake up and is a key parameter for controlling power consumption. If nodes wake up too frequently power is wasted initialising and powering down idle circuits, if nodes wake up too infrequently, delay is increased. RMAC is a novel MAC protocol that adapts a nodes frame length to traffic patterns. For networks in which traffic patterns are not known a priori, simulation results show RMAC saves significantly more power compared to HYMAC, a state of the art protocol with a prefixed frame length.

Keywords: WSN, MAC, Renewal theory, RL

1. **Introduction**

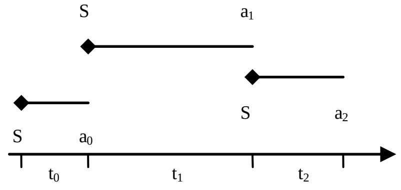
Periodic networks, (Wang & Akyildiz, (2009); Wang & Zhang, (2008)) are networks in which sensor values are transmitted to sink nodes deterministically and at fixed points in time. For this class of network, duty cycling, (Mihaylov et al (2012), Zhenzhen & Itamar, (2006); Ye et al, (2004), Udenze[a], (2014), Udenze[b], (2014)) where nodes alternate between active periods during which messages are exchanged, and sleep periods during which nodes go into low power states to conserve energy, has been proven to be an effective Media Access Control (MAC) strategy. In effect the MAC protocol design problem converges to finding optimal duty cycles, if nodes spend too long sleeping, delay is increased and if nodes spend too long awake, idle listening, (Mihaylov et al (2012); Udenze[a], (2014); Udenze[b], (2014)), energy is wasted. When traffic patterns are known a priori, optimal duty cycles can be determined and set. When traffic patterns are not known a priori, nodes equipped with a mechanism that adapt duty cycles to network conditions have been proven to improve on static pre-tuned cycles in terms of power conservation, (Mihaylov et al (2012); Udenze[a], (2014)). Reinforcement Learning (RL) is one such mechanism, (Sutton & Barto (1998)).

RL based agents find optimal actions by interacting with the environment and observing the consequences of actions taken in the form of rewards. This behaviour makes them well suited to MAC protocol design in which network conditions are not known a priori, (Mihaylov et al (2012)). Actions taken at decision epochs for the RL based MAC agent being how long to spend in the active and sleep states at the beginning of each time frame thereby forming a discrete time decision process. Modelling the MAC decision process as a Discrete Time Markov Decision Process (DTMDP), (Puterman, (1994)) and using adequate reward update mechanisms such as Q learning, (Sutton & Barto, (1998)), optimality can be proven for the resulting RL based agent's policies. Results for DTMDP based MAC agent protocols can be found in Mihaylov et al (2012), Zhenzhen & Itamar, (2006); Ye et al, (2004), Udenze[a], (2014), Udenze[b], (2014).

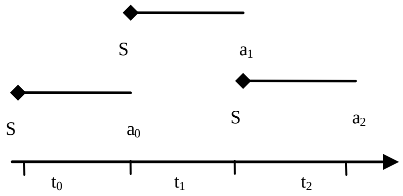
Figure 1 shows a time line for a DTMDP. Duty frames are fixed a priori and nodes wake up at the beginning of each frame to decide on how long to stay active before going into a sleep state for the rest of the frame. The fixed frame length in effect determines the frequency at which nodes wake up. This makes it an important parameter for a duty cycled MAC protocol. If the frame length is too short nodes wake up too frequently and if the frame length is too long, unnecessary delay is incurred. The process of waking up and initializing the nodes RF circuits as well as other circuits including the processor use up considerable amounts of energy. In Santos et al (2008), the power consumed shutting down circuits at sleep periods and waking up at active periods is quoted as up to twice the power consumed in the active period. In Torres (2006), power consumed waking up and going to sleep is quoted as 150mW and duration of 10ms. This amounts to a significant waste of power if no messages are transmitted.

RMAC is a protocol designed for networks in which traffic conditions are not known a priori and therefore frame lengths cannot be set a priori. The RMAC agent is based on Renewal theory (Nebres (2011); Sigman (2009)), a continuous time process which extends the discrete time analyses of afore mentioned works. A RMAC agent makes decisions on how long to spend in the sleep state depending on the rate of traffic. After a sleep period, the node enters a ready state or an active state depending on the state of the queue. If the queue is empty, it remains in the ready state until a message arrives else is transitions to the active state where it remains until the queue is empty before transitioning to an idle state. The time line for the RMAC agent is presented in figure 1. It can be seen that the RMAC agent adapts a nodes frame length to network conditions thus nodes spend the optimal amount of time in sleep and active states conserving energy that would be wasted waking up at each frame as in HYMAC, Udenze[a] (2014).

Figure 1: Event driven versus discrete time analysis



Continuous time process



Discrete time process

2. Literature review

WSN traffic patterns can be broadly classified as periodic and event driven, Wang & Akyildiz, (2009); Wang & Zhang, (2008). Periodic networks tend to be networks in which sensor values are transmitted to sink nodes at deterministic points in time; an example being collecting data from the environment, (Martinez et al (2005); Steere et al (2000)). For this type of network traffic patterns are frequently modelled by exponential distributions and the resulting queuing process treated as a Poisson process, (Mihaylov et al (2012), Zhenzhen & Itamar, (2006) and Udenze[a] (2014)). MAC protocols for periodic networks is presented as a duty cycling exercise in Mihaylov et al (2012), Zhenzhen & Itamar, (2006) and Udenze[a] (2014). The work presented in Ye et al (2004), SMAC, assumes that network traffic is known a priori such that duty cycles are preset. The works in, Dam & Langendoen, (2003) and Zheng et al (2005) present improvements to SMAC for traffic conditions that are dynamic but again known a priori. Where traffic conditions are not known a priori, optimal duty cycles have to be learnt online. RL has been proven to be successful at learning optimal actions by interacting with a given environment and taking actions that maximise long term rewards. Sutton & Barto (1998) present an in-depth study of RL algorithms.

RLMAC, Zhenzhen & Itamar, (2006), is a RL based MAC protocol that adapts a nodes duty cycle to network traffic conditions by interacting with the network environment online. DECMAC, Mihaylov et al (2012), is another RL based MAC that adapts to traffic conditions online by synchronising and desynchronising node's wake up times so that collisions are reduced. HYMAC, Udenze[a] (2014) combines the network traffic adapting capabilities of RLMAC with the collision avoidance mechanism of DECMAC to form a new protocol that adapts to varying traffic conditions and avoids collisions thus improving on energy efficiency compared to DECMAC and RLMAC.

A MDP, Putterman (1994), is one in which the current state of the system depends only on the previous state of the system and the action taken. It is shown in, Sutton & Barto (1998) that Q learning is a RL learning protocol that converges on optimal policies for MDPs. DECMAC, RLMAC and HYMAC all assume a DTMDP as the underlying system process such that optimality can be proven for resulting RL policies.

While HYMAC improves on DECMAC and RLMAC, it has a flaw in that the frame lengths are set a priori. WSN node power models presented in Torres (2006) show that power used in transitioning a node to a sleep state and waking up is approximately 150mW with a total transition time of approximately 10ms for a Mica mote. Compared to the energy used in transmitting a bit of information which is quoted as 3.12µJ (Torres (2006)), it is clear to see that this is a significant amount of energy. Thus where frame lengths are not optimal, a significant amount of energy is wasted going into and coming out of sleep states. RMAC is designed to put nodes in sleep states for long enough to minimise this energy wastage. In contrast to the DTMDP process of HYMAC, DECMAC and RLMAC, RMAC is based on a continuous time process, Renewal theory and Renewal reward theory. Furthermore, using an appropriate RL learning algorithm, in this case Q learning, policies can be proven to be optimal. For brevity, an overview of Renewal theory and Renewal reward theory is presented next.

2.2 Renewal theory

Renewal theory may be expressed by Arrival epochs or Renewal times, Inter-arrival times or a Counting process {N(t):t≥0} where N(t) is the number of arrivals to a system in interval (0,t]. $\bar{X}$ and E[X] are the mean inter-arrival time and expected mean inter-arrival time respectively. Σ^2 is the variance of the inter-arrival interval.

We are interested in the behaviour of the system as t→∞. Let N(t)/t be the time average renewal rate over the interval (0,t].

1.) The strong law of large numbers for renewal theory states that {N(t)/t;t>0} has a limit

$$\lim_{t\to\infty}\frac{N(t)}{t}=\frac{1}{\bar{X}} \qquad (1)$$

2.) Elementary renewal theorem states that

$$\lim_{t\to\infty}E[N(t)/t]=\frac{1}{\bar{X}} \qquad (2)$$

Where E[N(t)/t] is the expected number of renewals in time t

3.) Blackwell's theorem states that for appropriate values of δ, the expected number of renewals in an interval (t, t+δ] approaches $\delta/\bar{X}$ as t→∞.

2.3 Renewal reward theorem

For a positive recurrent renewal process in which a reward R_j is earned during cycle length X_j and such that $\{(X_j;\ R_j):j\geq1\}$ is an independent identical distribution with $E[R_j]<\infty$, the long run rate at which rewards are earned is given by

$$\lim_{t\to\infty}\frac{R(t)}{t}=\frac{E(R)}{E(X)}\text{ with probability 1} \qquad (3)$$

Also, $\lim_{t\to\infty}\frac{E\left(R(t)\right)}{t}=\frac{E(R)}{E(X)}$ (4)

A more in-depth study of renewal theory can be found in Cox (1970).

3. RMAC

Renewal theory generalises the Poisson process by allowing for any independent identical distribution, IID, thus a G/G/1 system is assumed. Assume transmitter events at a sensor node by which is meant received and transmitted packets is suitably modelled by a general IID. The time taken to process the packets is also suitably modelled by a general IID. The queue only advances when the transmitter is idle. Thus the queue state of the sensor node follows a G/G/1 queuing system. Further, of particular interest is the state of the queue when no events are pending, at this state the system is probabilistically the same and the cycle starts again thus is a renewal system and forms a renewal process.

Assume that the queue empties at times t_n, $n \geq 1$ forming a renewal process with IID inter-arrival times $x_n = t_n - t_{n-1}$; $n \geq 1$ ($t_0 = 0$). Suppose R_j denotes the jth cost of the events within the renewal period (incurred in the processing of the transmitter events that took place within the renewal period X_n)

Letting

$R(t) = \sum_{j=1}^{N(t)} R_j$ = total amount incurred by time t, where N(t) is the counting process of the renewal process.

Of interest is

$$\lim_{t \to \infty} \frac{R(t)}{t} \qquad (5)$$

Assume that pairs of positive (X_j, R_j) are IID thus R_j is allowed to depend on the length X_j but not on other lengths.

R(t) can be written as $\frac{N(t)}{t} \times \frac{1}{N(t)} \sum_{j=1}^{N(t)}$ (6)

From the strong law of large numbers (1) and elementary renewal theorem (2)

$\lim_{t \to \infty} \frac{R(t)}{t} = \frac{E(R)}{E(X)}$ with probability 1. (7)

Also $\lim_{t \to \infty} \frac{E(R(t))}{t} = \frac{E(R)}{E(X)}$ (8)

The rate at which rewards are earned is equal to the expected reward over a cycle divided by an expected cycle length.

The rate at which costs are incurred in terms of power and delay is equal to the expected cost over a cycle divided by an expected cycle length.

3.2 RMAC reward

One technique to solving the problem would be to observe transmitter events, build a suitable model and then solve the ensuing optimisation problem using Linear Programming (Putterman (1994)) or any such technique. In the absence of models, RL offers a solution. RL based 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 RMAC RL agent reward is as follows.

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 and made a ratio of the sleep time *t*. The weight *w* in equation 12 controls the desirability of long sleep periods, longer delays lower power, over short ones, shorter delays, higher power.

$$\hat{Q} = \sum_{j=0}^{t} Q_{s+j}^{t} \qquad (9)$$

$$Q = \hat{Q}^{w}/_{t} \qquad (10)$$

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)

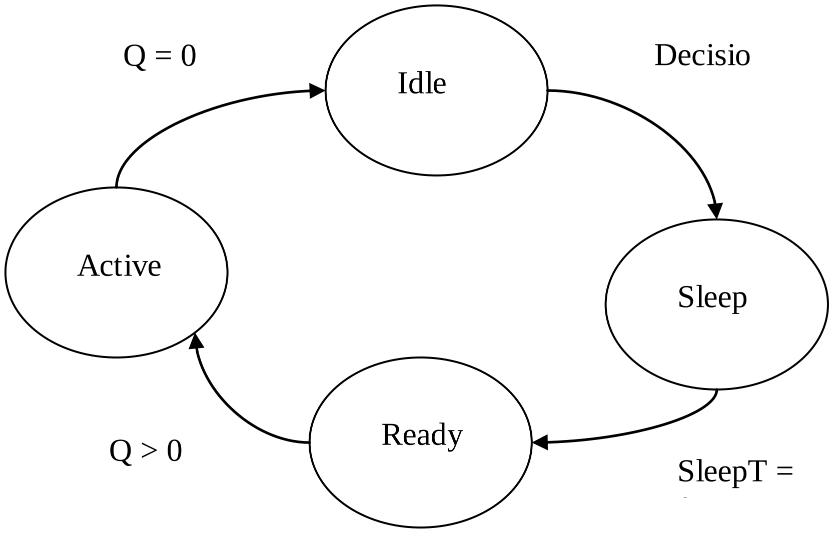
$$Q_{k+1}(s,a) = \begin{cases} Q_k(s,a) + \alpha\delta & if\ s_k = s,\ \ a_k = a \\ Q_k(s,a) & otherwise \end{cases} \qquad (11)$$

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

3.3 RMAC controller

Assume that the network is synchronised periodically during which time new nodes establish communication with neighbours. In the active state, with radios turned on, nodes attempt to exchange messages. If a node's queue is empty it listens for incoming messages otherwise it attempts to transmit. A listening node will time out after a set period and transition into an idle state if no messages are detected and the queue is empty. A node in the active state carries on exchanging messages until the queue is empty, after a set timeout period the node transitions into an idle state where the process is renewed. In the idle state a node makes a decision on how long to sleep for. The decision is communicated to all nodes before the node transitions to a sleep state. In remains in the sleep state for the set time after which it transitions into a ready state where it remains until an event where after it transitions to the active state. Figure 2 shows a state transition diagram for RMAC.

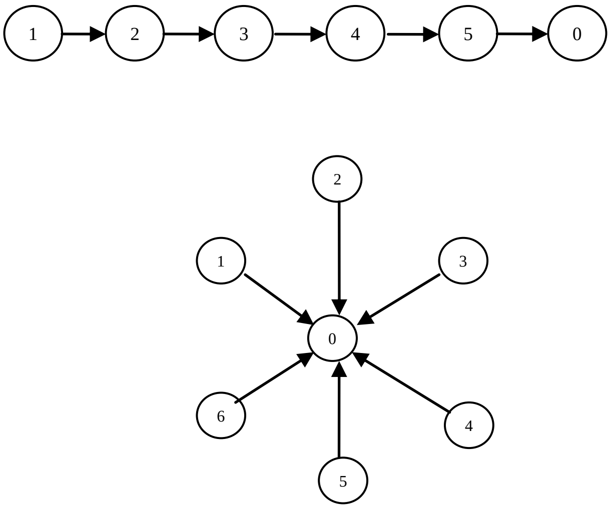
Figure 2: RMAC transition diagram



4. Simulation results

To determine the efficacy of the RMAC protocol, the following simulations were carried out. HYMAC is a state of the art protocol that combines the collision avoidance mechanism of DECMAC with the dynamic properties of RLMAC and was therefore used as a benchmark. Two topologies were simulated, a linear network consisting of 5 nodes and a sink node; and a star topology consisting of 5 nodes and a sink node, figure 2.

Figure 3: Simulated topologies, linear and star



For the HYMAC protocol, the frame length was set as 1sec and each frame split into 100 slots of 10 ms. The power used in initializing circuits after a sleep period was set to 75mW and the time taken to initialize set to 5ms. To power down circuits power is consumed at 75mW and the time to power down set to 5ms. A packet contains 50bytes and takes 20ms to transmit. The learning rate for the RL algorithm was set to 0.1 and the weight *w* in equation 10 set to 1.1, each simulation was run for 5000s. Simulations results presented below are averages over 50 runs.

Figure 4 shows the delay results for the linear topology with message inter-arrival times set to between 0.5 and 5s. Both the HYMAC and RMAC protocols perform similarly at low inter-arrival times with RMAC showing a slight increase in delay of 5%. This is because at low inter-arrival times the 1s frame length of HYMAC is optimal. As inter-arrival times increase, RMAC shows and increase in delay of 11% compared to HYMAC. This increase in delay the author puts down to the increased sleep durations and longer frame lengths of RMAC. The power consumption results of figure 5 however show that there is a significant reduction in power for RMAC compared to HYMAC. At low inter-arrival times, the reduction in power is 7% but quickly rises as the inter-arrival times increase. At 2.5s inter-arrival time the reduction in power is 46% for RMAC compared to HYMAC and at 5s the reduction in power is 79%. The author puts the performance of RMAC down to the following factors: 1.) Nodes have longer sleep times as inter-arrival time increases, saving significant amounts of energy in initializing and shutting down idle components. 2.) Nodes wake up at staggered times thus reducing collisions and energy wasted in retransmissions. 3.) Due to the event driven nature of RMAC, nodes spend just the right amount of time in the sleep state saving energy wasted during the exploration phase of HYMAC.

Next a star topology was simulated. Figure 6 shows the delay results for the star topology. At low inter-arrival times, HYMAC outperforms RMAC slightly with an 8% reduction in delay. At longer inter-arrival times, the margin increases to 11%. The power results of figure 7 however show a significant reduction in power for RMAC compared to HYMAC. At low inter-arrival times the different is 11% and increases to 71% at higher inter-arrival times.

Figure 4: Linear topology average delay

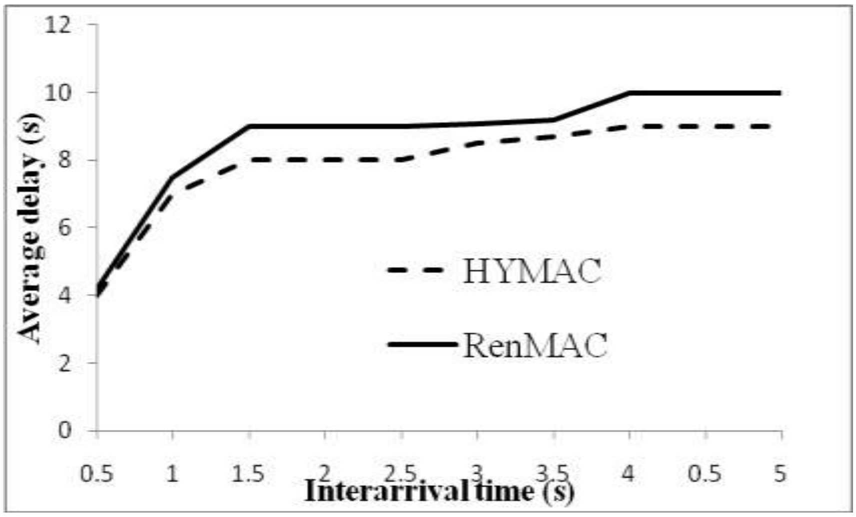


Figure 5: Linear topology average power

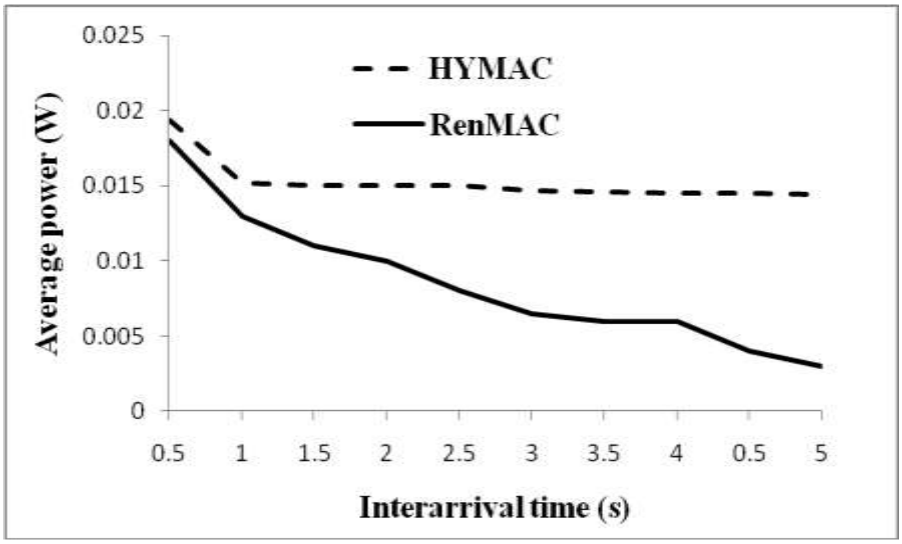


Figure 6: Star topology average delay

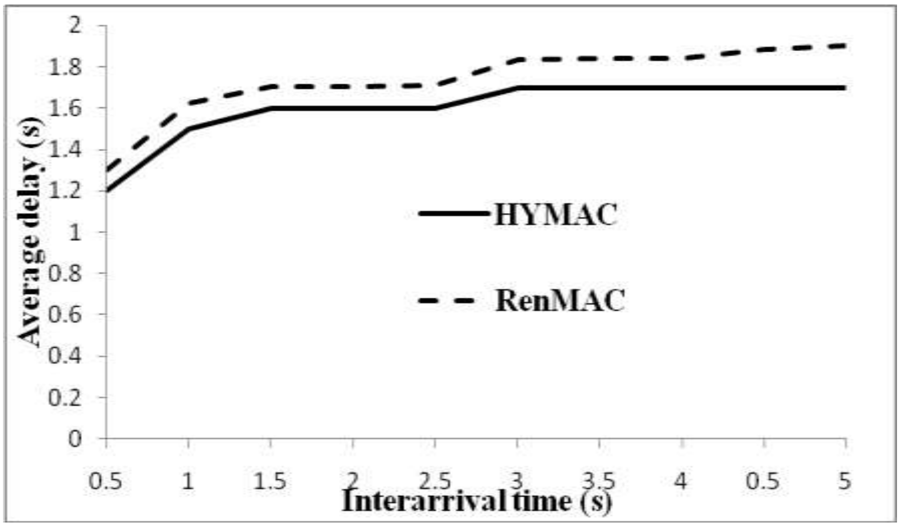
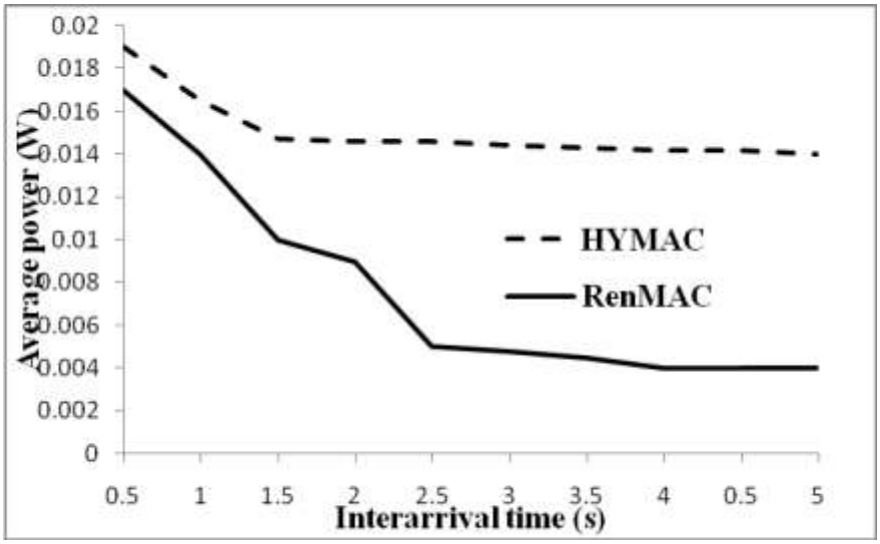


Figure 7: Star topology power



Conclusions

Prefixed frame lengths in duty cycled WSNs where traffic patterns are not known a priori can lead to suboptimal power management where nodes expend energy unnecessarily initializing and powering down idle circuits. RMAC is a MAC protocol designed to adapt a nodes duty cycle frame length to its traffic patterns. Simulation results have shown that RMAC outperforms HYMAC a state of the art MAC protocol in terms of power savings against increase in delay. RMAC was able to reduce power by as much as 70% for long inter-arrival times with only an 11% increase in delay. The simulation results were for traffic patterns modelled as an exponential distribution, in future the author will be experimenting on other independent identical distributions.

References

[1] Cox, D. (1970), “Renwal theory”, London: Methuen and Co.

[2] Dam, T.V. and Langendoen, K. (2003) ‘An adaptive energy efficient MAC protocol for wireless sensor networks’, SenSys ’03: Proceedings of the first International Conference on Embedded Networked Sensor Systems, New York, NY: ACM Press, pp.171–180.

[3] Martinez, K., Padhy, P., Riddoch, A., Ong, H. L. R., and Hart, J. K. (2005), “Glacial environment monitoring using sensor networks,” in Proc. ofWorkshop on Real-World Wireless Sensor Networks REALWSN’05.

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

[5] Nebres, P. (2011), “Renewal theory and its applications”, www.math.uchicago.edu.

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

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

[8] Santos, A. F., Marques, M. O., Bonatti, I. S. (2008), “Power aware model for wireless sensor nodes, dt.fee.unicamp.br.

[9] Sigman, K. (2009), “Notes on the Poisson process”, IEOR 6711.

[10] Steere, D, Baptista, A., McNamee, D., Pu, C. and Walpole, J. (2000), “Research challenges in environmental observation and forecasting systems,” in Proc. of 6th International Conference on Mobile Computing and Networking (MOBICOMM’00), pp. 292–299.

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

[12] Torres, M. G. C. (2006), “Energy consumption in wireless sensor networks using GSP”, d-scholarship.pitt.edu.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] Zheng, T., Radhakrishnan, S. and Sarangan, V. (2005) ‘Pmac: an adaptive energy-efficient MAC protocol for wireless sensor networks’, Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS’05), pp.65–72.

[20] 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