

Renewal theory sleep time optimisation for scheduling events in Wireless Sensor Networks

*Adrian Udenze and Klaus McDonald-Maier
Department of Computing and Electronic Systems,
University of Essex,
Wivenhoe Park, Colchester, CO4 3SQ, UK.
audenz@essex.ac.uk, kdm@essex.ac.uk.*

Abstract

This paper addresses the problem of optimised decision making in scheduling non deterministic events for WSN nodes. Scheduling events for highly constrained WSN nodes with finite resources can significantly increase the lifetime of the network. Optimising the scheduling of events ensures that under any given constraint the network lifetime is maximised. The presented technique uses Renewal theory to formulate a stochastic decision making process. By observing network events, optimised decisions are made regarding node sleep times. This technique links the time a node spends in the sleep state to the rate of traffic throughput in the network making the process able to adapt to changes. The proposed technique also has the added advantage of using data available locally to a node thus minimising control overheads. It can be employed in both static and ad hoc networks, as well as for autonomous decision making in nodes that have to self configure. Finally, this policy driven technique exploits the heterogeneous nature of a typical WSN architecture by using less constrained nodes for formulating policies which can then be implemented in more constrained nodes. Theoretical and empirical results are presented.

1 Introduction

Wireless Sensor Networks (WSNs) of small, light weight and low cost network elements (often called Pico nodes) that may be deployed in large numbers have significant potential for monitoring so called smart environments on earth as well as numerous applications in space. One of the challenges for achieving the potential of this technology is the effective management of the power consumption in these devices in order to maximise node lifespan and ultimately network lifespan while at the same time maintaining adequate quality and quantity of service. Power management of power constrained electronic

systems has received much attention and there are several paradigms that address this problem [1-4]. With the finite resources of these Pico nodes, power management can be thought of as maximising available power for given performance constraints. In other words, decisions will have to be made on how much power to expend for a given task. In a dynamic environment, this will have to be repeated to suit changes. This decision making process is relatively straightforward in deterministic networks where data is collected at fixed points in time. For non deterministic networks, where events are random and there is a constraint on delay i.e. data is required as and when they happen, the decision making problem becomes more complex. Further more, to maximise the lifespan of the nodes, any decisions made should ideally be optimal. Scheduling network activity and operation offers significant energy savings since by knowing when events and hence. packets transmissions will occur, power can be saved by transferring idle nodes into sleep mode as well as ensuring that media access is fair, thus minimising retransmissions due to collisions. In a deterministic environment, deciding on a schedule is tractable and the decision can be made to guarantee optimality. In a non deterministic environment stochastic processes offer a technique to deal with the random nature of events. The stochastic decision making strategy used in this paper is based on Renewal theory. It forms the basis for making optimised scheduling decisions. This technique fundamentally analyses the network as having two commodities to be managed, network traffic and energy consumption. Managing these commodities involves making decisions and carrying out actions. Ideally, every decision and subsequent action should be best for a desired outcome. This technique also offers the potential for reducing scheduling overheads. A policy is a prescribed set of actions for a given goal over a period of time. A given policy remains valid for as long as parameters remain stationary otherwise indefinitely. Policies can also be formed by any network node with adequate resources and implemented in other nodes. The problem formulation fundamentally makes a link between

network traffic and node activity, consequently scheduling is dynamic. This has the added benefit of generating policies that do not depend on specific network configurations but simply on traffic throughput, further reducing any overheads. A further advantage of the technique presented is the potential for autonomous decision making. The decision on how long to sleep for is taken based on information available to a node i.e. traffic flow and service rates. By analysing this information, a node can determine an optimum policy without consulting other nodes for information. This has the advantage of further reducing overheads and also makes the technique suitable for nodes that have to self configure intelligently and without any human intervention.

The remainder of this paper is structured as follows – first a review of related work is presented followed by a formulation of the scheduling problem in section 3. Then a theoretical basis for the scheduling problem is described in section 4 before a solution is given, followed by a simulation of a typical event based system. Finally conclusions are drawn and future work set out.

2 Related work

Where information is required from a WSN at fixed points in time i.e. taking temperature readings from a plant every 1 second, the network events can be thought of as deterministic. For non deterministic networks, events are random and importantly there is a constraint on the delay incurred in getting data from source to sink. Without a delay constraint, for the example above, readings could still be made at the same fixed periods thus network is deterministic [15]. Consequently any scheduling technique for these networks fall broadly into one of these two categories. Deterministic scheduling techniques are presented in [6-9]. [6-8] present Time Division Multiple Access (TDMA) scheduling where timeslots are allocated by receivers and filled by transmitters. A crucial consideration is how well a given technique scales as network size increases. Centralised techniques as in [6] do not scale well whereas distributed ones [7] tend to introduce significant overheads. The technique presented in [8] follows on the work done in [7] which was designed predominantly for ad hoc networks in which data is collected at tree roots (sense to gateway applications). Based on required bandwidths, parents advertise available slots and children respond to these advertisements depending on their needs. This extension to the original work allows for two way communication, partial flows of traffic in which the flow terminates at a node and not necessarily at the root, as well as better synchronisation.

Another deterministic scheduling technique is presented in [9]. The authors describe the merits of the protocol to include power savings by eliminating idle listening, avoids packet collisions which reduces packet delay with

very little buffering required. On the down side, this scheduling protocol requires a deterministic environment and does not attempt any load balancing. Further examples of power reduction by way of deterministic scheduling can be found in [16-19].

For mission critical applications and where delay overheads are important, the use of these techniques may not be adequate and instead non deterministic techniques should be employed. One such technique is presented in [10]. This technique focuses on maintaining a network “backbone” that ensures packets can be forwarded from a source to a destination at a given time. Each node in the network makes a decision periodically on whether to stay awake and act as a coordinator or to go to sleep. As this technique uses only local information, it scales well as number of nodes increases. In [11] the IEEE 802.11 protocol for power management in ad-hoc networks is presented. In this scheme time is divided into beacon intervals. At the beginning of each beacon interval there exists a time interval called the ATIM window (Ad-hoc Traffic Message Indication Message Window) where every node is awake. When a node has a message to transmit it first transmits an ATIM frame to the intended destination during the ATIM window which in turn replies with an ATIM-ACK acknowledgement. A node that did not receive or transmit an ATIM frame during the ATIM window can go to sleep at the end of the ATIM window. Clearly the size of the ATIM window will have a substantial influence on the performance of the power saved. [12-13] discuss the influence of the ATIM window on the power saved. As shown in [13], the optimal ATIM window size depends on various factors and a mechanism to dynamically choose the ATIM window size is given. The work presented in [14] improves on the IEEE 802.11 protocol by eliminating the ATIM window and replacing it with the DATA window which reduces overheads.

In summary, for data critical systems in non deterministic environments, any improvements in delay must be carefully weighed against power consumption. The IEEE 802.11 is a typical example where the time spent between waking up has to be balanced with delays in forwarding packets. Stochastic processes have been used to solve similar decision making problems. In [23,24,29, and 30], Markov chains are used. Queuing theory [26,27] and Renewal theory [30] also closely related to Markov have also been applied. Such work forms the basis for making optimised scheduling decisions in this paper. In particular Renewal theory is used to formulate and optimise node sleep times with a view to guaranteeing optimal delays for a given power constraint.

4.1 Renewal theory

Renewal theory [20-22] began as the study of stochastic problems concerned with the failure and replacement of components. Assume a population of components, each of

which is characterised by a non negative continuous time random variable X called its failure time. Suppose that the random variable has a probability distribution function (p.d.f.) $f(x)$, zero for negative x , that is

$$f(x) = \lim_{\Delta x \rightarrow 0^+} \frac{\text{prob}(x < X \leq x + \Delta x)}{\Delta x} \quad (1)$$

$$\text{With } \int_0^{\infty} f(x) dx = 1 \quad (2)$$

Suppose that at time zero a new component begins life. This component fails at time X_1 and is replaced immediately by a new component with failure time X_2 . Then the second failure will occur after $X_1 + X_2$ and likewise if the failure time of the r th component used is X_r , then the r th failure occurs at time

$$S_r = X_1 + X_2 + \dots + X_r \quad (3)$$

If $\{X_1, X_2, \dots\}$ are independently identically distributed random variables, all with p.d.f. $f(x)$, the system is an ordinary renewal process. Other forms of renewal processes (modified renewal process) can be found in [21].

The cumulative distribution function (c.d.f.), $F(x)$ giving the probability that a component has failed by time x is

$$F(x) = \text{prob}(X \leq x) \quad (4)$$

$$= \int_0^x f(u) du \quad (5)$$

Note that (5) gives $F(x)$ in terms of the p.d.f. $f(x)$ and conversely, differentiating (5) gives

$$f(x) = F'(x) \quad (6)$$

Determining $f(x)$ for a given $F(x)$.

While $F(x)$ above gives the c.d.f. of failure time, the survivor function $F_s(x)$,

$$F_s(x) = \text{prob}(X > x) \quad (7)$$

$$= 1 - F(x) \quad (8)$$

$$= \int_x^{\infty} f(u) du \quad (9)$$

gives the probability that a component has not failed up to time x . Of particular interest is the forward recurrence time V_t which is defined as the time measured from t to the next renewal to occur after t . Other parameters of interest in the study of renewal theory can be found in [21]. This is sometimes referred to as the residual life-time. It can be proved that if t is large, the forward recurrence time V_t in an ordinary renewal process has the p.d.f.

$$V_t = F_s(x)/u \quad (10)$$

where $F_s(x)$ is the survival function (9) and u is the mean failure time. Finally, a renewal process in which X_t has the p.d.f. $F_s(x)/u$ is called an equilibrium renewal process. Thus an equilibrium renewal process can be regarded as

an ordinary renewal process in which the system has been running a long time before it is first observed.

3 The scheduling problem

There are many applications for WSN and consequently the network structure can vary considerably for different applications. Often the network structure tends to be heterogeneous with high level less resource constrained nodes acting as gateways or data sinks for other more constrained nodes [5]. The technique presented here takes advantage of this heterogeneous nature of the network by using tethered nodes to analyse network traffic based on which a power management policy is derived. This is in turn broadcast to the more constrained nodes for implementation. This technique divulges the resource intensive task of formulating a policy from highly constrained nodes. Also, as the system is policy driven, a policy once formulated and broadcast for implementation lasts for a long time, thus eliminating communications overheads for the duration of the policy. Other considerations to be taken into account when designing protocols for these highly constrained networks include whether or not sources and sinks are stationary. When the nodes are mobile, ad hoc networks, the amount of traffic going through a node has an added dimension of uncertainty rather than in a static environment. A scheduling strategy involves synchronising communication between two or more nodes. When the environment is dynamic this synchronisation process will have to be performed continuously. This increases the overheads associated with the strategy and has a negative impact on network performance. The technique presented here works on traffic flow through a node abstracting the number or identity of nodes sending the information, this helps to reduce the number of schedule updates due to roaming. A further goal for the formulation of a scheduling strategy for these highly constrained nodes is the requirement for any strategy to be straightforward. This is also achieved as constrained nodes require little resources by way of computing power or memory for implementing this policy driven approach. Note that, while the approach presented here requires input from an operator, it has the potential to be adopted for autonomous implementation. Figure (1) below shows three nodes – A, B and C. All nodes can communicate with each other. Nodes B and C use node A as a gateway to a base station situated further up the network hierarchy. Nodes B and C act in turn as routers for other nodes lower the network hierarchy. All eventually arrive at a base station which has an infinite amount of resources and is used for policy formulation. In an ad hoc environment nodes are mobile and the figure 1 represents a snapshot of communicating nodes at a given time. The total traffic arriving in each node, for example (A) is

$$A_T = X_A + \sum_i X_{iA} \quad (11)$$

where X_{iA} is the traffic arriving from node i to A and X_A is the traffic originating in A , with $i = [B, C, \dots]$. In this instance (with three nodes A, B, C) on each node (A) the total flow of traffic into the node is 2 (B and C) and $i = 2$ plus the traffic generated from the sensors on the node (A).

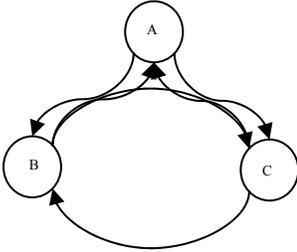


Figure 1: Communicating nodes

Ideally what is required is for node A to wake up just in time to service requests from nodes B and C , events generated in node A can trigger an interrupt to wake the node up. Because the events in all nodes are random, determining the optimal time to wake up and service requests is further complicated. However, by observing the rate of request arrivals from both nodes, in theory A can determine when the next event is likely to occur and wake up in time to service it. Consequently A is setting up a schedule using only information available to A without any need to communicate with nodes B and C . Also, as A considers the average rate of arrivals irrespective of where the data is coming from, then if nodes B and C migrate, and in turn, other nodes take their places, any policy formed remains valid so long as average rate of arrivals remain the same. Following is a problem formulation and solution for optimum sleep times for node A . For the purpose of analysing the renewal process formed, assume that the service rate and request arrival rates follow a Poisson process. This is a fair assumption as if arrivals at nodes B and C follow Poisson distributions, then the total traffic in A is the sum of Poisson processes which is also Poisson. In this case, the rate at which events are serviced have no dependence on any previous requests and is totally random. Thus the system is said to be memory less and the probability of completing a service follows the exponential distribution. This class of systems is referred to in queuing theory as an $M/M/1$ system [26-27]. Other variations include systems with general probability distributions ($G/M/1$ or $M/G/1$) which renewal theory is also applicable for, however assume a classical $M/M/1$ system for the purpose of the analysis done here. Note also that it is not necessary to repeat the process for every node (or groups of nodes)

in the network as all nodes communicating with the same network properties can use the same policies. Assume that the rate at which the service processor (node A) services requests follows an exponential distribution with mean λ_d and the service arrivals have a mean of λ_u . That is the probability of an arrival or request service both have the form:

$$F(t) = 1 - e^{-\lambda t} \quad (12)$$

From (2) above, $F(x) = F(t)$. The system nodes can be in one of four states as shown in figure 2 below namely active, sleeping, transitioning to sleep or transitioning to active. Note that the analysis here can be extended to systems with more states such as alternative low power states. Upon completion of a service request, the system transitions into the sleep state, where a decision is made on how long to go to sleep for. Following this decision, the system goes to sleep. After the set sleep time the system wakes up and transitions into the active state, where it waits for and services the next request that arrives. After this service it transitions back into the sleep state and the cycle begins anew, which forms a renewal process. Note that in this case the component being renewed is the arrival and subsequent servicing of an event in the network. Also, consider a queuing process in which service is available only at service intervals, a renewal process. A customer arriving at time t will have to wait a time V_t for the first service instant. Also, observation of the system is delayed for a long time such that the renewal process is an equilibrium renewal process. Thus the problem of finding the optimal schedule is equivalent to minimising V_t the time spent waiting for service. Further more, the p.d.f. of V_t for an equilibrium renewal process is given in (10) above. Thus the problem reduces to finding and minimising $F_s(x)/u$ (10). Assume that the action set available to the power manager is to sleep for jh where j is an index and h is a fraction of the arrival time. Each action has an associated cost / reward. The reward for going to sleep for jh time period is a function of the rate at which power is saved sleeping. Assume $q(j)$ the average performance penalty calculated by averaging the time penalty a service request incurs over $t(j)$ the renewal time. The expected energy $\sum p(j)e(j)$ is calculated using $p(j)$ the probability of issuing a command to go to sleep and $e(j)$ the expected energy consumption. The optimisation problem is to find the optimal distribution of the random variable $\Gamma = [0, h, 2h, \dots, jh, \dots, Nh]$ (where N is the maximum sleep time for a given arrival rate) that specifies how long to spend in the sleep state. The solution can be viewed as a table of probabilities with each element $p(j)$ specifying the probability of spending jh time in the sleep state. The solution can be found by using the results of the ratio limit theorem for renewal processes [22] as shown below.

$$\begin{aligned}
& \min \frac{\sum_j p(j)q(j)}{\sum_j p(j)t(j)} \\
& \text{s.t. } \sum_j p(j)[e(j) - t(j)P_{\text{constraint}}] = 0 \quad (13) \\
& \sum_j p(j) = 1 \\
& p(j) \geq 0 \forall j
\end{aligned}$$

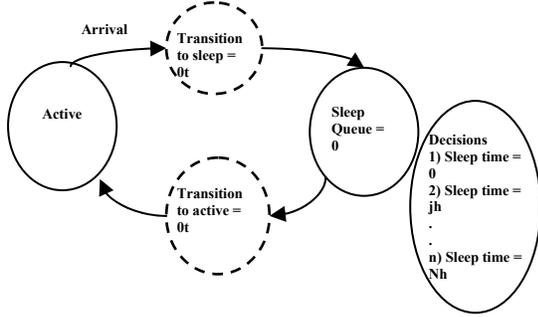


Figure 2: State transitions

$q(j)$ above is a function of the residual life as defined in (7) and (10) and u is equal to $t(j)$ the distribution of which is given in (1). The expression sum of $p(j)$ equals one comes from (2).

4 Theoretical analyses

Ultimately the aim is to find the cost associated with taking each action. The set of actions is a random variable spread over Γ . Further more the renewal time and subsequent residual life is a function of each action. The rate of request arrivals follow a Poisson process with rate λ_u and the servicing time of requests also follow a Poisson process with rate λ_d . Let the mean time until arrival equal to β and let $I(j)$ be an indicator function which is equal to one if $j < N$ and zero otherwise. The expected time until renewal can now be defined as

$$\begin{aligned}
t(j) = & E[t(j)I(\beta \leq jh)|\Gamma = jh] + \\
& E[t(j)I(\beta > jh)|\Gamma = jh] \quad (14)
\end{aligned}$$

Which is the sum of the expected time until renewal if the arrival time is less than or equal to the sleep time, in addition to the expected time for an arrival after the sleep time.

Observing figure 2 above, for an arrival during sleep time, the estimated renewal time is

$$\begin{aligned}
& E[\tau_j I(\beta \leq jh)|\Gamma = jh] = \\
& \left[jh + EU_1 + EU_2 + jh \frac{\lambda_d}{\lambda_d - \lambda_u} + EU_1 \frac{\lambda_d}{\lambda_d - \lambda_u} \right] P(\beta \leq jh) \quad (15)
\end{aligned}$$

and for an arrival after the sleep period –

$$\begin{aligned}
& E[\tau_j I(\beta > jh)|\Gamma = jh] = \\
& \left[jh + EU_1 + EU_2 + \lambda_u + \frac{1}{\lambda_d - \lambda_u} \right] P(\beta > jh) \quad (16)
\end{aligned}$$

that is, the expected time until renewal $t(j)$ is equal to the probability of arrival in the sleep time times the total time taken to sleep, make transitions and service incoming requests, plus the probability of arrival after sleep time times the rate of arrival plus the time taken to service requests and make transitions. The probability of an arrival interval jh is given in (17) below.

$$P(\beta \leq jh) = \int_0^{jh} f(x) \quad (17)$$

The costs can now be calculated as follows -

i.) *Delay cost* : This is calculated from equations (15) and (16) above by simply replacing the sleep time (jh) with the average delay incurred for an arrival during sleep time and zero for an arrival after sleep time. Hence the delay in the active state is simply the time spent waiting for an arrival plus the time taken to service the request when it arrives.

ii.) *Power cost* : Again power cost is calculated by as for the delay cost however the time spent in each state is multiplied by the rate of power consumption in that state. An example is provided in the results section below. Queuing systems with vacation have been studied in [31] where average delay due to vacation time Vac_t is given as

$$\frac{(Vac_t)^2}{2 * Vac_t} \quad (18)$$

4.1 Policy implementation

The optimal policy obtained from solving the LP derived from (13) is a table of probabilities $p(j)$ which is transformed into a table of cumulative probabilities $P(j)$ based on $p(j)$. The sequence of sleep time actions at an intended node follows this probability distribution. For example, from figure 1, node A sleep policy might be to sleep for 30 min with a probability of 0.4 and 60 min with probability 0.6. The sequence of actions for the next 10 events is simply to choose action 30 min to action 60 min with a ratio of 4 to 6. Upon deciding the order of these actions, node A informs nodes B and C of the schedule. For ad hoc networks, this information can be communicated during initialisation or simply by employing a random timeout until an initial

communication is made following which the schedule is established.

5 Results

For the purpose of demonstration it is assumed that the arrival rate of requests is 1 every hour and the service rate of requests when the system is active is 1 every minute. Therefore the service time and arrival rate models are

$$F_d(t) = 1 - e^{-t} \text{ and } F_u(t) = 1 - e^{-0.0166t}.$$

Assume that the system consumes power at a rate of 10mW awake [25] and 0.01 mW sleeping, as well as a lump sum cost of 10mW to wake up and 5mW to sleep. In the sleep state a small amount of power is assumed to be consumed keeping certain circuits like timers active. Transition delays are deemed small enough to be ignored. Furthermore, it is assumed that the action set available to the power manager is to go to sleep for 0min, 60min, 120min and 180min. The maximum sleep time (180min) is set to approximate the maximum time before an arrival, i.e. probability of an event arrival after 180min is 1.

Example: For a sleep time of 60min, *Expected renewal time* $E(t(j)) =$

$$j = 1; h = 60; EU_1 = 0; EU_2 = 0;$$

$$\frac{1}{\lambda_d - \lambda_u} = 1.02; P(\beta \leq 60) = 0.632;$$

$$P(\beta > 60) = 0.368; \lambda_u = 60; Avg_delay = 30$$

$$\therefore t(j) = (60 + 0 + 0 + 1.02)0.632 +$$

$$(60 + 0 + 0 + 60 + 1.02)0.328 * 1.02 = 78.26$$

Expected delay $E(q(j)) =$

$$j = 1; h = 60; EU_1 = 0; EU_2 = 0;$$

$$\frac{1}{\lambda_d - \lambda_u} = 1.02; P(\beta \leq 60) = 0.632;$$

$$P(\beta > 60) = 0.368; \lambda_u = 60; Avg_delay = 30$$

$$\therefore t(j) = 30 * 0.632 + 0 + 0 + 0.328 * 1.02 = 19.29$$

Expected energy $E(E(j)) =$

$$j = 1; h = 60; EU_1 = 10; EU_2 = 5;$$

$$\frac{1}{\lambda_d - \lambda_u} = 1.02; P(\beta \leq 60) = 0.632;$$

$$P(\beta > 60) = 0.368; \lambda_u = 60; pow_active = 10mW$$

$$pow_sleep = 0.01mW$$

$$\therefore e(j) = (60 * 0.01 + 10 + 5 + 1.02 * 10)0.632 +$$

$$(60 * 0.01 + 10 + 5 + 60 * 10 + 1.02 * 10)0.328 = 221.57mJ$$

Similarly the expected renewal time, performance and power penalties were calculated yielding the table (1) below.

Table 1: Theoretical results

Sleep time decision (min)	Expected Renewal time (mins)	Expected Energy (mJ)	Expected Power (mW)	Expected Perf. penalty (mins)
0	61.02	610.2	10	1.02
60	78.26	221.57	2.83	19.29
120	129.14	97.4	0.82	53.92
180	185.93	33.81	0.18	92.95

```
f = [1.02 19.29 53.92 92.95];
Aeq = [610.2 221.57 97.4 33.81 ;
        61.02 78.26 129.14 185.93];
beq = [5; 1];
lb = zeros(4,1);
[x,fval,exitflag,output,lambda] =
linprog(f, [], [], Aeq, beq, lb, []);
```

Figure 3: Matlab LP file

The minimisation problem described in (16) cannot be solved directly but has to be transformed into a linear programming (LP) problem using intermediate variables.

$$\text{Let } y(j) = \left[\frac{p(j)}{\sum_j p(j)t(j)} \right]; z(j) = \frac{1}{\sum_j p(j)t(j)} \quad (19)$$

then LP

$$\begin{aligned} \min & \sum_j q(j)y(j) \\ \text{s.t.} & \sum_j [e(j)y(j) - t(j)z(j)P_{constraint}] = 0 \\ & \sum_j t(j)y(j) = 1 \\ & z \geq 0 \end{aligned} \quad (20)$$

Proof of (19) and (20) is given in [28]. Once the intermediate variables $y(j)$ and $z(j)$ are obtained from the LP, the probability of going to sleep for a jh period is given as $p(j) = y(j)/z(j)$.

As can be seen in table 1, the two parameters of interest in the network are now completely tractable. The power consumption of the node can be set to any value from the maximum 10mW to a minimum of 0.18mW. Alternatively the delay incurred by a node can be set from 1.02 min to a maximum of 92.95. The LP problem stated above can also be used to find appropriate commands for a given constraint, e.g. for a maximum power consumption of 5mW, the LP (figure 3) is solved to give issue a command of sleep time = 0min with probability 0.36 and sleep time = 60 with probability 0.64, incurred delay is 12.71min. Thus decisions can now be made in a manner that guarantees optimality and the system operator knows exactly how much delay is incurred for each action. For example, setting the sleep time to 60 min yields a delay of 19.29 min which is not optimal if a delay

of 40 min is adequate, thus power is wasted. A graph of delay against power consumption is plotted in figure 4 below. An M/M/1 queuing system was also simulated and delay and power consumption measured to compare to the theoretical results. Figure 5 shows the delay incurred for each sleep period in seconds and figure 6 illustrates the power consumed in mW for each action. Figure 7 shows Poisson generated events, figure 8 shows Poisson generated service rates and figure 9 shows state transition times for an action to sleep for 1 hour (3600 sec). To reduce the number of data points required for figures 7, 8 and 9, simulation was ran at 3 second time steps. The simulation results agree with theoretical results.

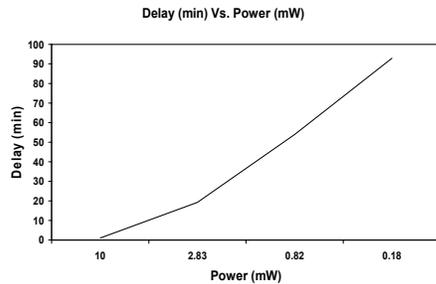


Figure 4: Average delay vs. average power

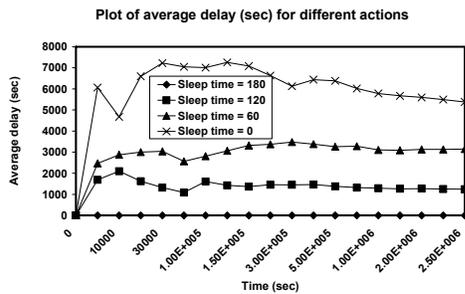


Figure 5: Average delay for different actions

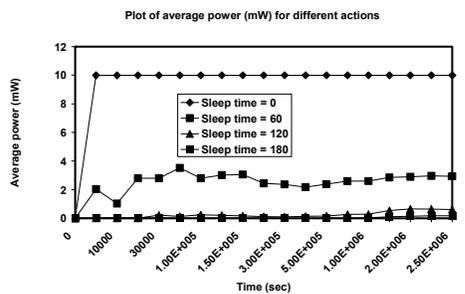


Figure 6: Average power for different actions

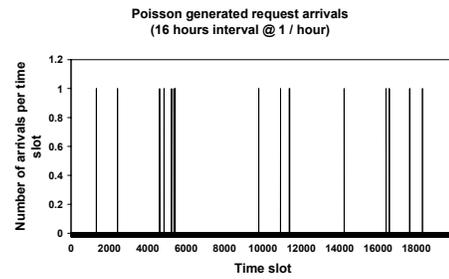


Figure 7: Poisson generated request arrivals

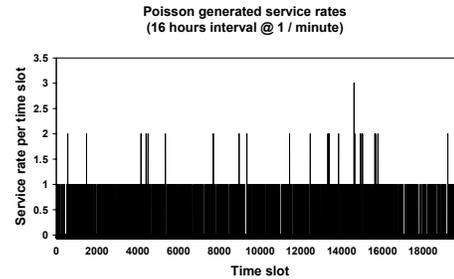


Figure 8: Poisson generated service rates

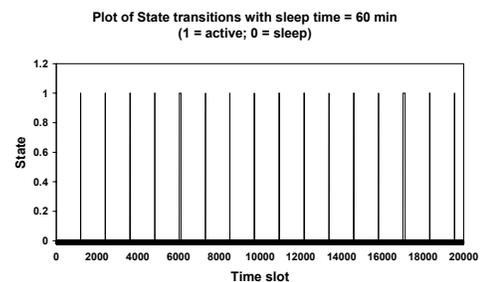


Figure 9: State transition times

6 Conclusions

This paper has investigated the problem of optimised scheduling decisions in WSN networks. For ultra low power nodes in a WSN, it is essential to maximise node life span so that any decisions made are optimal. Further more, networks operating in a non deterministic environment present added challenges due to the random nature of events. Continuous time renewal theory was used to formulate and solve for an optimised sleep time policy. A policy once formed remains valid for as long as network parameters remain the same and in a stationary environment, can be proved to be optimal. Secondly, the technique presented here takes advantage of the heterogeneous nature of resources in the network by formulating resource intensive policies in less constrained nodes for implementation in more constrained nodes.

Thirdly, the scheduling decisions are made based on information local to a node without the need to communicate with other nodes. All of these help to reduce the communication overheads needed for a schedule. Finally, for networks that have to self configure, it is essential that nodes have a mechanism for making intelligent decisions in non deterministic environments and for choosing the best options for any given operating condition. The presented technique has the potential to be used in such systems. Hidden Markov chains have been used extensively in artificial intelligence and the problem of autonomous actions for intelligent networks can be considered to fall within this realm. To this end future work will involve extending the work for application in self organising networks. Research into autonomous agents operating in non deterministic environments will be undertaken.

Acknowledgements

This research is supported by the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) under Grants EP/C005686/1 and EP/C54630X/1.

References

- [1] G. P. Halkes, T. Van Dam and K. G. Langendoen, "Comparing energy-saving MAC protocols for Wireless Sensor Networks", *Mobile Networks and Applications* 10, 783-791, 2005, pages 783-791.
- [2] N. Bambos, "Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues and Design Aspects", *IEEE Personal Communications*, June 1998, pages 50-59.
- [3] A. Sinha, A. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks", *IEEE Design and Test of Computers*, March-April 2001, pages 62-74.
- [4] J. Hill, R. Szewczyk, et al., "System Architecture Directions for Networked Sensors", In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, Apr. 2000
- [5] W. Hu, C. T. Chou, S. Jha, N. Bulusu, Deploying long-lived and cost-effective hybrid sensor networks, in: *Proceedings of The First Workshop on Broadband Advanced Sensor Networks. (BaseNets 2004)*, San Jose, CA, October 25, 2004
- [6] K.A. Arisha, M.A. Youssef, M.F. Younis, "Energy-aware TDMA based MAC for sensor networks," *IEEE IMPACCT*, 2002, New York City, NY, USA, May 2002.
- [7] Barbara Hohlt, Lance Doherty, Eric Brewer, "Flexible power scheduling for sensor networks", *Proceedings of the third international symposium on Information processing in sensor networks*, Pages: 205 - 214 , 2004
- [8] B. Hohlt, A Brewer, "Twinkle: Network power scheduling in sensor networks", *CSD-05-1409*, September 12, 2005
- [9] M. L. Sichitiu "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks" *IEEE Infocom*, 2004.
- [10] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: An efficient coordination algorithm for topology maintenance in ad-hoc wireless networks" *Wireless Networks*, Volume 8 , Issue 5, September 2002, pages 481 – 494, 2002
- [11] The editors of IEEE 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, 1997
- [12] E.-S. Jung and N. H. Vaidya, Energy Efficient MAC Protocol for Wireless LANs," in *INFOCOM 2002*, June 2002
- [13] H. Woesner, J.-P. Ebert, M. Schlager, and A. Wolisz, "Power-Saving Mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective," *IEEE Personal Communications*, June 1998
- [14] E.-S. Jung and N. Vaidya, "A power saving mac protocol for wireless networks," *Technical Report, UIUC*, July 2002
- [15] O. Chipara, C. Lu, and G. Roman, "Efficient power management based on application timing semantics for wireless sensor networks", *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05) - Volume 00*, Pages: 361 – 370, 2005
- [16] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, "MACAW: A media access protocol for wireless LANs", *Proceedings of the conference on Communications architectures, protocols and applications*, London, Pages: 212 – 225, 1994
- [17] C. L. Fullmer, J. J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks", *Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication*, France, Pages: 39 – 49, 1997
- [18] C. Wu, V. Li, "Receiver initiated busy tone multiple access in packet radio networks" *Proceedings of the ACM workshop on Frontiers in computer communications technology*, US, Pages: 336 – 342, 1987
- [19] Phil Karn, "MACA - A new channel access method for packet radio", *ARRL 9th Computer Networking Conference. ARRL*, 1990
- [20] H. Taylor and S. Karlin, *An introduction to stochastic modelling*, New York: Academic, 1998
- [21] D. R. Cox, "Renewal theory", London: Methuen & Co. ltd, Cat no. Methuen, 2/5238/11
- [22] S. Ross, *Stochastic processes*. New York: Wiley 1996
- [23] W. J. Andreson, "Continuous time Markov Chains: An application oriented approach", New York Springer, 1991
- [24] Y. George, "Continuous time Markov chains and applications: a singular perturbation approach", New York Springer, 1998
- [25] CC 1000. Chipcon Corporation. CC1000 low power FSK transceiver. <http://www.chipcon.com>, Dec. 2006
- [26] R. L. Dsiney and P. C. Kiessler, "Traffic processes in queuing networks : a Markov renewal approach", *Johns Hopkins University Press*, c1987, ISBN 8950 0157 8
- [27] M. H. van Hoorn, "Algorithms and approximations for queuing systems", Amsterdam: Centrum voor Wiskunde en informatica, c1984, ISBN - 8730 1358 1
- [28] S. Boyd, "Convex optimisation", *Stanford Univ. Stanford, CA, Stanford Class Notes*, 2007
- [29] G. A. Paleologo, L. Benini, A. Bogliolo, G. De Micheli, "Policy Optimisation for Dynamic Power Management", *IEEE trans. CAD of Integrated Circuits and Systems*, Vol. 18, no.6, pp. 813-833, June 1999.
- [30] T. Simunic, L. Benini, "Event driven power management", *IEEE Trans. On CAD if IC and Sys.*, Vol. 20, No. 7, July 2001.
- [31] N. G. Chee Hock, "Queueing modelling fundamentals", *John Wiley & Sons*, ISBN 0471968196.