ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 10(31), DOI: 10.17485/ijst/2017/v10i31/113873, August 2017*

Prediction based Hard Disk Power Conservation Algorithm in Mobile Ad hoc Networks using APRIORI

Amirul Islam* and Amanpreet Kaur

Yamuna Institute of Engineering and Technology, Gadholi – 133103, Haryana, India; amirulislam464@gmail.com, aman_preet_k@yahoo.co.in

Abstract

Objectives: The main objectives of this work are to enable mobile nodes predict the switch off states of Hard Disk and other electronic components using APRIORI algorithm which has pre-mined list of timings of every request to be serviced by the node. **Methods/Statistical Analysis:** The application of this Algorithm is shown via Simulink (MATLAB). We use a Sequence generator of variable inputs (on/off) signals which controls the load circuit of the simulated node. The input of the sequence generator is defined by an example depicting a Mobile Node traversal and represented by a Table residing in the memory of the node. The table values are mined using preset values of Confidence and Support as in APRIORI algorithm. The data set used in the simulation represents all cases of the Algorithm in real time. **Findings**: Controlling the switch off states using pre-mined data values based on frequent item sets enables a Mobile Node to effectively decide the switch off states or power saving states for various electronic hardware, which includes Hard Disk. The Main power drawn in Hard Disk is by the Motor of the cylinder which if turned off and on (Spin UP/Spin DOWN) effectively will result in more efficient battery savings for each Mobile Node as compared to Adaptive and Threshold based Spin UP/ Spin DOWN based policies. **Applications/Improvements:** Since the switch timings decided by the algorithm entirely depend on mined values, application of better data mining techniques will result in more efficient battery operation.

Keywords: Power Conservation, Prediction based Algorithm, Wireless Networks, APRIORI, MANETS

1. Introduction

Power conservation can be a tricky thing when it comes to preserving the working efficiency of a mobile node. MANETS already have a limited power source issue. The battery of each node needs to be charged again beyond a given threshold of SOC. We propose a prediction based model which can be very effective especially in those networks where the nodes follow the same set of timings or those networks with pattern based adaptive routing decisions[1]. For all service requests from the Cluster Head or the node generated requests, which is when the node acts as the source and Cluster Head as a sink. In both cases, we define the event as a service request from either of the side. The time of each event of request arrival at a particular node can be mined using the association rules as defined in the APRIORI approach, hence different from OPT (optimization) based approach. Many MANETS use or travel the same path or follow the same set of activities per routine or per cycle of battery discharge up to a certain SOC threshold value and are designed in such a way that the node lasts till the cycle of events is complete. The

*Author for correspondence

power conservation schemes in mobile ad hoc wireless networks can be divided into System based and Device based. The consumption of power source in a HDD is spin up, the power consumption in various hardware components of the mobile node is shown in the Figure 1, as in [2]. Since the approximate usage of the HDD is 20% that of the total power consumed by the node, the change in savings of the power in the battery is scaled to1/5th of the total savings. Although the CPU clock and the RAM need continuous power for system calculations. The power saving is hence mostly variable in other hardware of the system.

We can mine the time of each event pertaining to each service request at that particular node and the expected runtime of each request using APRIORI. We use the confidence_level (c) and support_level (s), mining association rules as in [3,4]. The mining process hence becomes a variable which is dependent on the usage and the personalized configuration of the mobile node. In this approach, a table is maintained based on a path the mobile node visits and the time of each event of a frequent service request using the pre-configured support_level and confidence_level is mined. Each node thus maintains a set of variables, e.g. for a Node A {confidence_level (c), support_level (s), soc_threshold (n), cycle_time(t), time_frame_length (t_f)}.

Here confidence_level is the minimum confidence level for a request based on the number of times or the frequency the node encounters the request, the support_level being the minimum support on which each event is considered, the soc_threshold being the battery percentage where the battery cycle ends and from where the node needs to be charged again, cycle_time is the time of the uptime of the node for the given soc_threshold, time_frame_length (t_f) being the integral time which represents the total cycle_time in multiple number of frames. These values can be altered and re-configured using variation in the support_level and confidence_level with time and

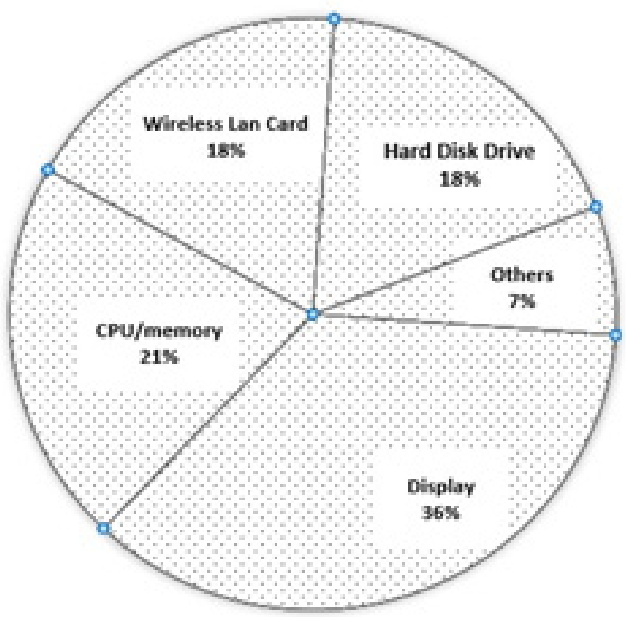

Figure 1. Power Consumption in a mobile ad hoc unit.

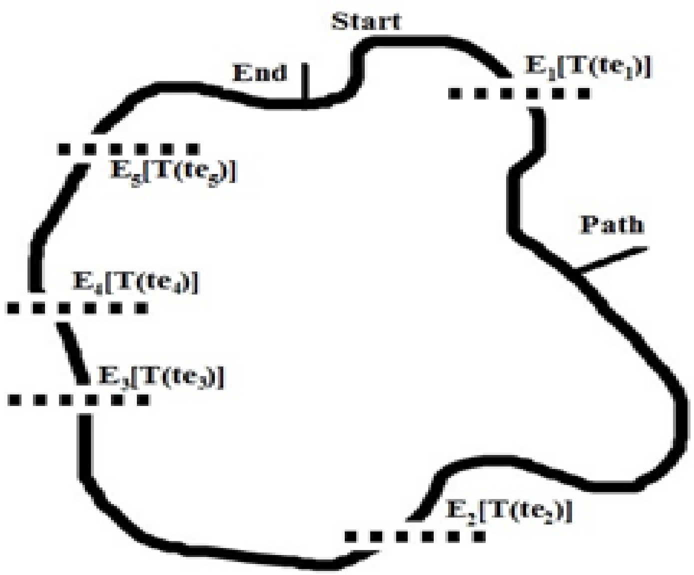

Figure 2. Path traversed by Mobile Node A and the experienced events of each request.

hence the efficiency along with power consumption will vary based on the configuration set by the user admin or Cluster head.

The time_frame_length (t_f) is also set based on average time taken by the node to process the requests. Now based on APRIORI the timing of each request event is known, so the HDD does not have to be spun down to a less RPM state, spun up from a dead/sleeping or a less RPM state. It can be completely turned off and when the next immediate request event approaches, the disk goes into disk_spinning state where the disk motor achieves maximum RPM which is at a time of event_time (t_n) minus spin_up_time (t_u), where the event_time being the time which has been mined from association rules via the APRIORI algorithm as in [4], and spin_up_time being the total time disk needs to attain maximum RPM. The expected time taken by the node to process the service_request (r) is also mined and defined as request_service_time (t_r).

It is understandable that when the HDD spins at its maximum speed the data read/write rates are expected to be efficient. After the request has expired the disk power is again switched off and disk is put in a disk_notspinning state where the HDD motor completely stops rotating and does not consume any power. The table is maintained by the Main memory of the node and now power is only consumed by the CPU clock and transmission systems e.g. LAN card, thus saving the total power consumed by the node. Figure 2 shows an example of a traversed path of a mobile Node A and each dotted line denotes the frequently mined time of each request arrival.

Each battery node traversal cycle is defined by the route it traverses and the set of events of service requests

Table 1. Event table of a mobile node

EVENT	Mined Event time. event_time (t_n)	Expected time of service for the request (t_r)	Disk enters disk_ spinning state	Frame Number
E_1	t_{e1}	t_{r1}	t_{e1} - t_u	2
E_2	t_{e2}	t_{r2}	t_{e2} - t_u	7
E_3	t_{e3}	t_{r3}	$t_{e3} - t_u$	10
E_4	t_{e4}	t_{r4}	$t_{e4} - t_u$	11
E_5	t_{e5}	t_{r5}	t_{e5} - t_u	14

it encounters. Table 1 shows the time of each event of request arrival or generation at node A and the time the disk enters the disk_spinning state which also includes spin_up_time. The table storage is maintained by the node itself, resides in the main memory and is maintained by the cluster head or a node admin. This table is used by the algorithm to determine the request timings and hence use it as a control signal for the HDD control unit to control the spin up and spin down signals of the motor. Since the node only is not responsible for the timing of each event, this algorithm hence uses a global clock for event synchronization[5,6].

2. Prediction based Management

Many different power management schemes as shown by[7,8], use the dynamic approach. Using APRIORI based predictive power management scheme we can save a higher percentage of battery in the mobile unit, although as stated above it requires computation and calculation of the timing of data arrival which will prompt a preemptive spin up and spin down of the disk. The spin down though can be variable, depending upon the time taken by the node to service the current request or whether if the node receives no request in the next time frame. Many power management schemes[8,9], use adaptive approach for spin down management. Using APRIORI based model the total cycle_time is divided into time frames, the disk will either continue to spin at the best possible speed or stop completely, and the time taken in spinning up is involved within that time frame. Since this method is completely prediction based so the efficiency of the node is completely dependent on how much accurate the unit is able to predict the arrival of request and the time for which the request has to be serviced. The length of the time frame is also decided in such a way that it shouldn't be too long, in that case the disk may continue spinning even when the request has already been serviced by the node in lesser time hence being inefficient or too short. In that case the disk might have to be spun up again which will result in low read/write rates because of frequent spin up and down cycles, which are in tens of thousands[10] in average cases. Even if the time_frame_length is set for shorter time intervals it has chances of being more efficient than longer time intervals, in an ideal scenario the time_frame_length would be more efficient if it is close to the average service time of the requests. That is if:

$$t_f = (\sum_{i=1}^{n} t_{r(i)})/n$$

Also, the time difference between two consecutive requests must not be less than the length of the time frame which will result in more frequent spin ups and switch of HDD states. The calculation of time frame for which the disk continues spinning is hence to be decided carefully and variably based on the usage of the node or the preference set by the Node Admin or Cluster Head.

3.1 Algorithm for Predictive on Demand Power Management

Step-1: The node sets a pre-defined constant which is time time_frame_length (t_f) and also maintains a table of each request_event and the time in which it is expected to be serviced. Runtime of each servive_request (r) may be variable, hence arrival time is only considered for calculation.

Step-2: The node checks in the mined event table of request arrivals the time of next immediate Request Event $t_{e(n+1)}$.

Step-3: As the time of each mined request arrives the node allocates a state of disk_spinning or a signal of (1) for that quantum equal to (t_f), to the disk, and the disk continues to spin for that time.

Step-4: If the service request is still present after the end of current disk_spinning session, the next frame state is also set to disk_spinning (1) time frame, till there is no request in the next frame.

Step-5: The node also checks in the table the time between the next expected request and the end of current service_request.

$$t_{e(n+1)} - [t_{e(n)} + t_{r(n)} + t_u] < t_f$$

And if the difference is less than time_frame_length, then the disk continues the state of disk_spinning for the next frame, else it enters disk_notspinning or a (0) state where the disk stops spinning and no power is drawn from the battery.

Step-6: The node repeats Step-2 to Step-5 till the total expected time of battery drain or when SOC equals soc_threshold or if total run time equals cycle_time.

The best case which is in terms of time of service requests and directly proportional to battery consumption of a given Service request (t_r), of this Algorithm is:

The worst case being:

$$P_{worst} = [t_f * a] + [2t_f * b] + [3t_f * \{n - (a + b)\}]$$

Where, a and b are the number of requests which occupy 1 and 2 number of frames respectively and rest number of requests are expected to occupy 3 frames. The average case is also in the bounds of the worst case.

3. Simulation and Results

Simulation of the above given Mobile Node A has been done using MATLAB, Simulink. This example is a representation of an average case of the Algorithm. Simulation is based on power conservation by the system, when it consumes the battery power. The simulation runs for 2600 seconds with the time_frame_length (t_f) of 160 with 15 time frames. The spin_up_time (t_u) is adjusted along with the time frames in the simulations. The input from the Event table is represented in the sequence in form of 0 and 1. Where 0 represents the frame where disk is in disk_notspinning, state and 1 representing when disk enters disk_spinning state, as shown in Figure 3. Each frame time represents a 1,0 input according to the state in which the disk is supposed to be.

The model uses a BLDC motor and a Li-Ion based battery of 12V and 6.5Ah capacity, as in a battery driven system[11]. The control of the circuit is done by using an Ideal Relay which is controlled by a Repeating Sequence Stair Generator, shown in Simulink block diagram Figure 3, the same simulation mode setup for a motor used in[12]. The Sequence Generator inputs are shown with time_frame_length (t_f) set as 160 seconds, as shown in Figure 3.

The discharge characteristics of the battery operating in all frames is shown in Figure 4, these discharge characteristics include all the hardware components of the mobile node. The rating of the DC motor is adjusted in

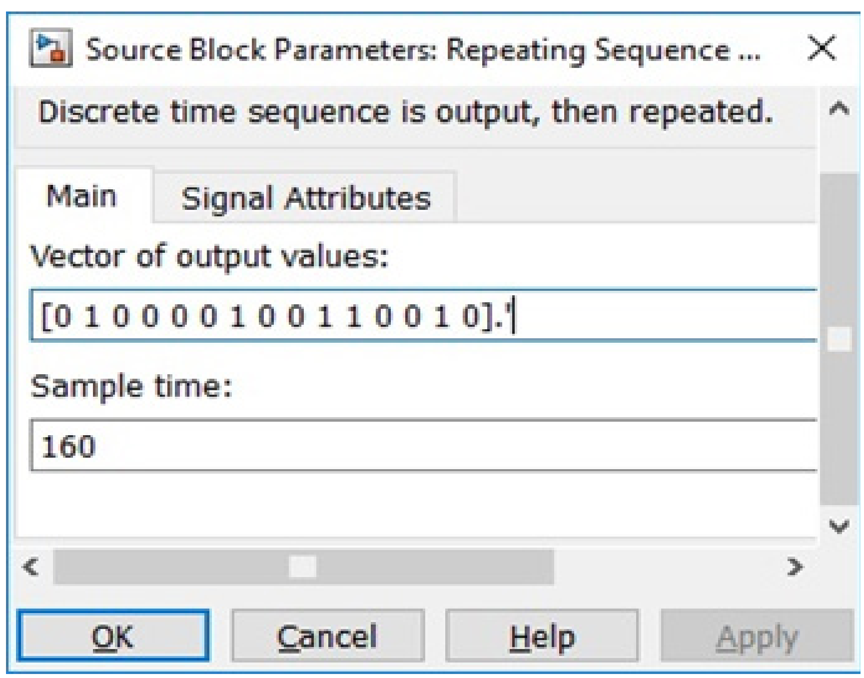

Figure 3. Input of the sequence control.

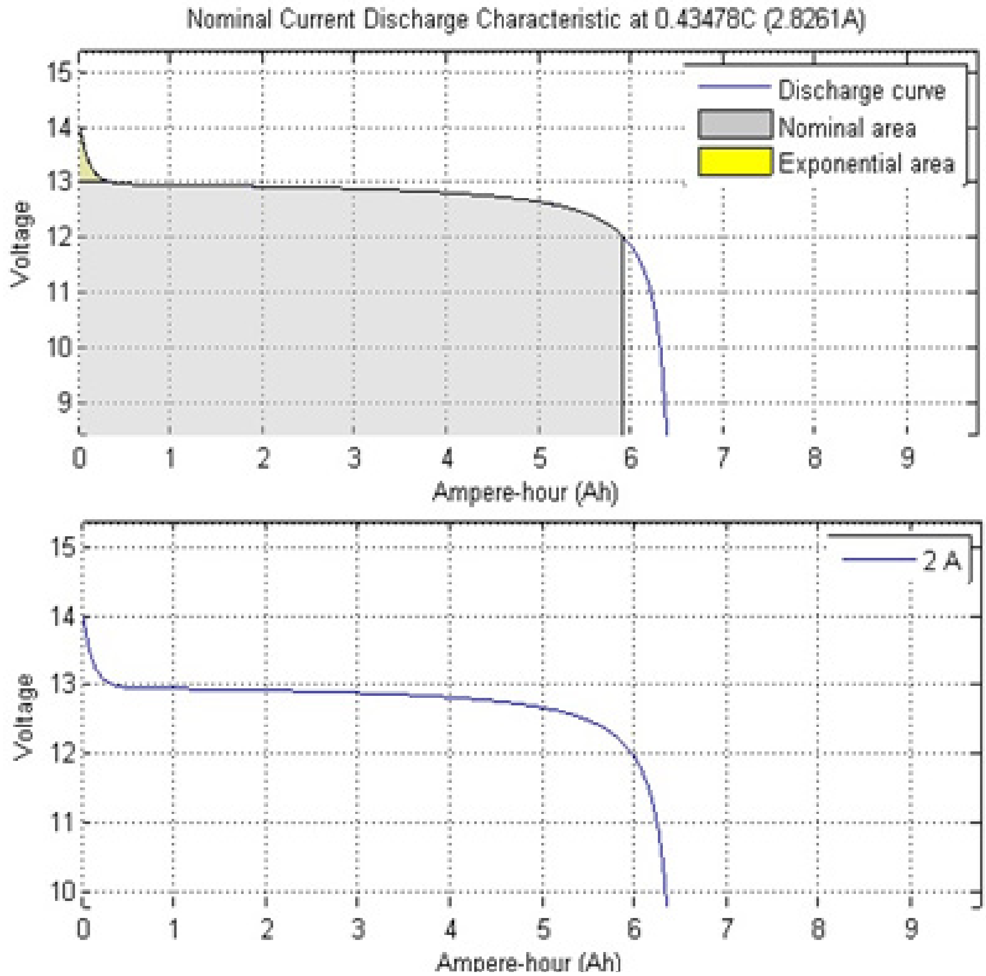

Figure 4. Discharge characteristics at 12V and 2A current.

such a way that it compensates for the absence of other hardware components. The mechanical load of the DC machine is a preset constant load in Newton/meter inclusive of weight of the HDD cylinder. The losses in noise and heat are also not considered in the load as they are negligible in MANET based nodes. The Li-Ion battery discharge and exponential area variations are also compensated by the control system in real time and are hence ignored in the simulation. The scope provides the result of simulation containing 4 graphs showing Voltage of the circuit, the SOC of the battery, Speed (RPM) of the HDD, the Armature current in the system on a similar scale of Time in (0-3000) seconds as shown in Figure 5.

The simulation shows a clear switch off state in the circuit whenever the Sequence control shows a value of 0 for a designated Time Frame. Clearly the SOC does not reach Zero for the same routine of 2600 seconds and the disk only enters a disk_spinning state and spins at 600 rad/s

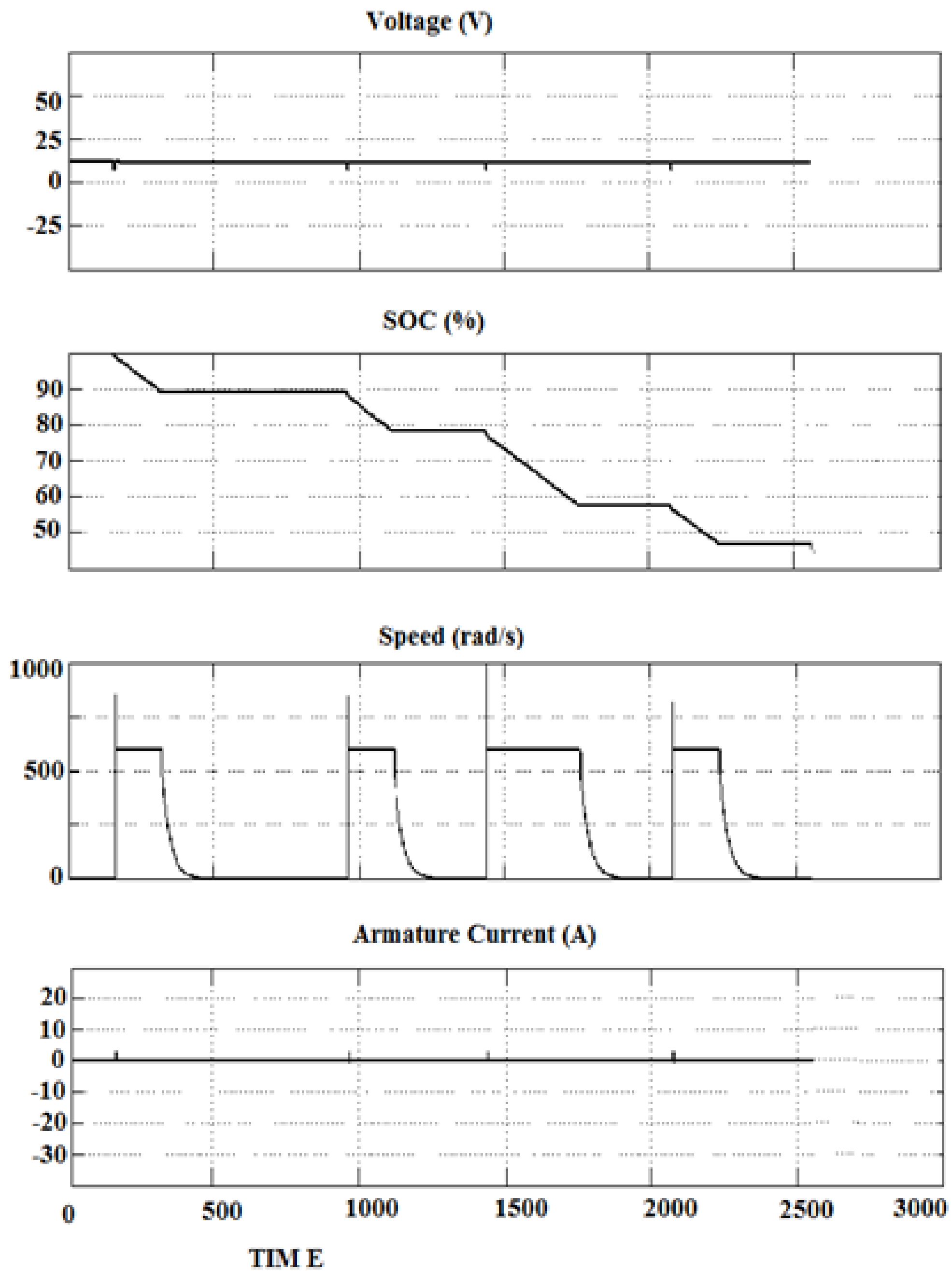

Figure 5. Operational battery attributes and speed control.

which is approximately 5700 RPM only when decided by the Algorithm based on mined data maintaining the same speed for that Time Frame.

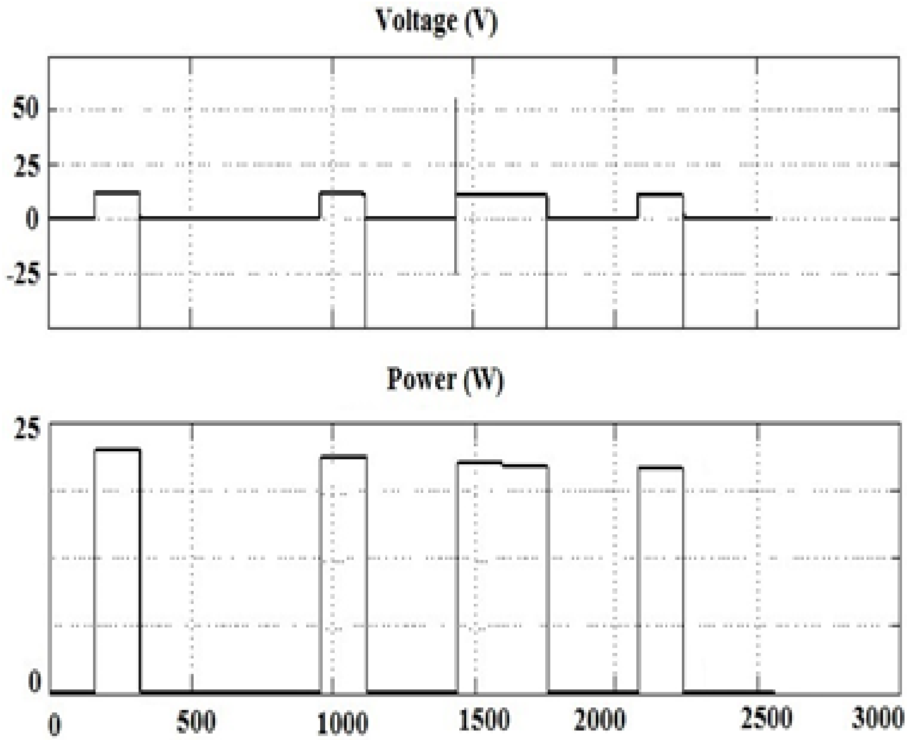

Figure 6. Switched power consumption by the load.

Figure 6 shows the circuit switch voltage and the power consumption by the load in the circuit. The simulation results in terms of power conservation can also be seen in the same figure, with Voltage peaks and Power Consumption peaks properly ending with the end of the designated frame. The disk hence does not keep spinning any further time if the frame containing a {0} signal in the Sequence Generator is found. The data vector of output values in Sequence Generator is taken from Table 1 of Node A. We can easily conclude the final power consumption timings of the real-time operation of Node A. The soc_threshold is not passed since a lot of power conservation takes place by switching off the motor of HDD completely. Only about 53% of the total battery was consumed in the whole operation.

4. Conclusion

Since the disk Spin up can also be request based or On Demand, the Spin down can be known using the APRIORI based prediction model for HDD, thus providing a better power saving method in MANETS. Although this approach may have limitations, only to be applicable in similar routine following models where we expect the mobile node to follow the same route and same set of timings of the requests along that route every cycle, the configuration of the variables in the algorithm also needs to be performed carefully based on the expected efficiency and Power saving ratio.

5. Acknowledgement

The work done with the simulation and results and the design of the paper text was done under the guidance of Asst. Prof. Amanpreet Department of Computer Science, and Prof. Manpreet Department of Electronics and Communication both from YIET Gadholi, Haryana, India.

6. References

1. Faizian P, Rahman MS, Mollah MA, Yuan X, Pakin s, Lang M. Traffic pattern-based adaptive routing for intra-group communication in dragonfly networks. 2016 IEEE 24th Annual Symposium on High-Performance Interconnects (HOTI), Santa Clara, CA; 2016. p. 19–26. Crossref.
2. Douglis F, Krishnan P, Marsh B. Thwarting the powerhungry disk. USENIX Winter Conference . p. 293–306. 3 . Chen Y, Du X, Zhou L. Transformer defect correlation analysis based on Apriori algorithm. 2016 IEEE International Conference on High Voltage Engineering and Application (ICHVE), Chengdu; 2016. p. 1–4. Crossref.
4. Ito Y, Kato S. An apriori-based approach to product placement in order picking. 2016 IEEE International Conference on Agents (ICA), Matsue, Japan; 2016. p. 114–15. Crossref.
5. Sasabe M, Takine T. Continuous-time analysis of the simple averaging scheme for global clock synchronization in sparsely populated MANETs. IEEE Journal on Selected Areas in Communications. 2013 Apr; 31(4):782–93. Crossref.
6. Scholz M, Bregulla F, Hinze A. Using physical clocks for replication in MANETs. Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07., White Plains, NY; 2007. p. 114–19. Crossref.
7. Paleologo GA, Benini L, Bogliolo A, De Micheli G. Policy optimization for dynamic power management. Proceedings 1998 Design and Automation Conference. 35th DAC. (Cat. No.98CH36175), San Francisco, CA, USA; 1998. p. 182–7. Crossref.
8. Douglis F, Krishnan P, Bershad B. Adaptive disk spindown policies for mobile computers. Computing Systems. 1995; 8:381–413.
9. Greenawalt PM. Modeling power management for hard disks. Proceedings of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Durham, NC; 1994. p. 62–6. Crossref.
10. Lu Y-H, De Micheli G. Adaptive hard disk power management on personal computers. Proceedings Ninth Great Lakes Symposium on VLSI, Ypsilanti, MI; 1999. p. 50–3. Crossref.
11. Lahiri K, Raghunathan A, Dey S, Panigrahi D. Battery-driven system design: a new frontier in low power design. Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15h International Conference on VLSI Design, Bangalore; 2002. p. 261–7. Crossref.
12. Lulhe AM, Date TN. A design & MATLAB simulation of motor drive used for electric vehicle. 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil; 2015. p. 739–43. Crossref.