

COMPARISON OF A PREDICTIVE AND A THRESHOLD BASED HARD DISK POWER MANAGEMENT SCHEME IN MANETS

Amirul Islam

Research Scholar, Department of Computer Science,
Yamuna Institute of Engineering and Technology, Jagadhri, Haryana-133103, India
amirulislam464@gmail.com

Amanpreet Kaur

Assistant Professor, Department of Computer Science,
Yamuna Institute of Engineering and Technology, Jagadhri, Haryana-133103, India
aman_preet_k@yahoo.co.in

Abstract - Power management in mobile units has always been a daunting task for designers and programmers. The Hard disk consumes a substantial amount of power resources available to the node in terms of Spin up and continuation of Disk spin. Many techniques and algorithms have been introduced under device power management schemes in various hardware components of mobile ad hoc nodes which are divided into two major types, Prediction based and Threshold based schemes. We compare a time based threshold technique with a preset time limit and an APRIORI based predictive scheme which uses data mining techniques. Then we compare and see the power saving efficiency of both schemes via simulation.

Keywords: MANETS; Ad hoc; Mobile Nodes; Threshold; APRIORI; Hard Disk.

1. Introduction

Power saving schemes in MANETS are based under Device Power Management and System software based schemes. Hard disk power conservation issues fall under Device Management. Since the node has a lot of hardware components which use the power available, out of that available power, Hard Disk uses about 20% [1]. The realistic power saving schemes for devices are Threshold based and Prediction based both relating to adaptive policies for Spin Up and Spin Down as in [2]. The main use of power in Hard Disks is the Disk Spin up, the longer the disk spins and the more frequently the control causes the disk to spin up, the more power is drawn from the battery source. Since a Service Request (r) may have any arrival time from a Sink in the network, the primary job of the Hard Disk power management scheme is to decide the Spin Down timing of the Disk. It is tempting to initiate the Spin down of the disk after the Service Request has ended, it may result in unnecessary Frequent Spin Ups in the disk which are a lot in number [5]. Resulting in the overall decrease in performance in terms of data read/write rates of the Node. The comparison Algorithms run till a time of minimum power level or any threshold limit is reached as in [4].

2. An APRIORI based power management Scheme

Many data mining techniques and tools available today enable us to perform frequent mining on various data sets available. In MANETS, the nodes receive the requests variably, and the Service request start and end timings can be mined via a set of association and mining rules decided by either the Node itself or the Cluster Head (CH). These mining methods as in [3][9] enable us to find and predict the average time of each service request or (t_r).

Consider a Node N receiving a set of requests from various sinks or self-generated source request in a given network, as in an On Demand request model [6][10]. The node maintains a set A which contains the timing values of all requests. These values are generated by the node or the Cluster head using Frequently mined data of Start and End time of each request. The data is stored in the Node memory with each node having its own data set. The mining metrics include use of Support and Confidence values as in [3] which are temporary to each node in the network and are controlled by the Cluster Head or the network Administrator. Hence each node defines a set say $A = \{t_{r1}, t_{r2}, t_{r3}, t_{r4}, \dots, t_{r(n)}\}$, Where $t_{r(n)}$ is the mined value of the runtime of the n^{th} request the node receives.

We also define t_f as the average time of each request in Eq. (1).

$$t_f = (\sum_{i=0}^n t_{r(i)})/2n \tag{1}$$

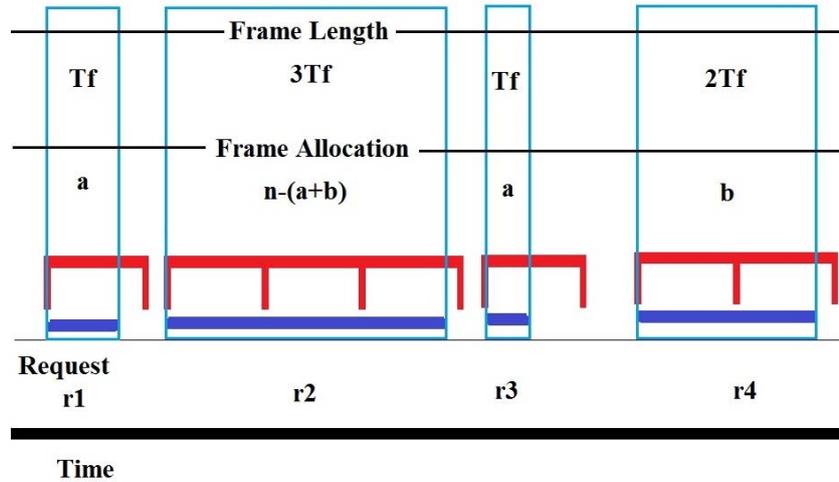


Fig. 1. Frame allocation method to Spin up of the Disk.

The half of the average time t_f is then set as a quantized frame, at the start of which the node constantly checks if and new request or pending request exists. In case a new request arrives, the node allocates a time frame equal to t_f to this disk spin up. And when the current allocated time frame ends the node checks whether the request is still present or has ended. If the $t_{r(n)}$ of that request exceeds the t_f , the node allocates another time frame equal to t_f to the disk spin, this process continues till all requests end or the battery runs out.

Consider the classification of the types of requests on the basis of the time it takes to complete. Where we have “a” number of requests which take only one Time frame to complete, and “b” as the number of requests which take two frames to complete, the requests which take more than 2 frames are $n-(a+b)$. As shown in Fig. 1. The frames allocated to “a”, “b” and “ $n-(a+b)$ ” number of requests is shown by Eq. (2). Which also represents the total uptime of the Hard Disk in seconds, also represented by red lines in Fig. 1. The requests which take more than 2 frames can be reduced to the sum of “a” and “b” type frames for simplification purposes. If not, we assume that requests which take more than 2 frames don’t take more than 3 frames of disk spin. Or the time taken by requests of “ $n-(a+b)$ ” number equals $3*t_f$.

2.1. Runtime Cases

The Worst and average cases are of the same bounds and are represented by Eq. (2), and best case of the total Uptime of the disk is represented by Eq. (3). The total disk uptime is represented by T_u .

$$T_u = \{a * t_f + 2 * b * t_f + 3 * n(a + b) * t_f\} \tag{2}$$

$$T_u = n * t_f \tag{3}$$

3. A Time based Threshold power management Scheme

In a time based threshold policy we use a preset time limit of (t_h) after the end of current request during which if the node receives no request the disk goes into a lower RPM spinning state or a state where the disk completely stops spinning as in [7]. This is different from a scheme by [6] where we calculate the node switch off time for charge recovery. For simplification in comparison we use the state in which the disk completely stops spinning. In case the node does receive a request in the given threshold time it continues spinning at highest RPM till the end of that request and then allocates another time limit for which the disk continues spinning waiting for another request. This policy is ideal in situation where we try to avoid the spin up time and power consumed by the disk in spin up from a sleep or dead state, which consumes a lot of power resources [5]. The worst case scenario of this policy is when each request doesn’t arrive in any of the next allotted threshold time. The disk has to be spun up again from a dead state each time a new request arrives. As shown in Fig. 2 with request r_2 .

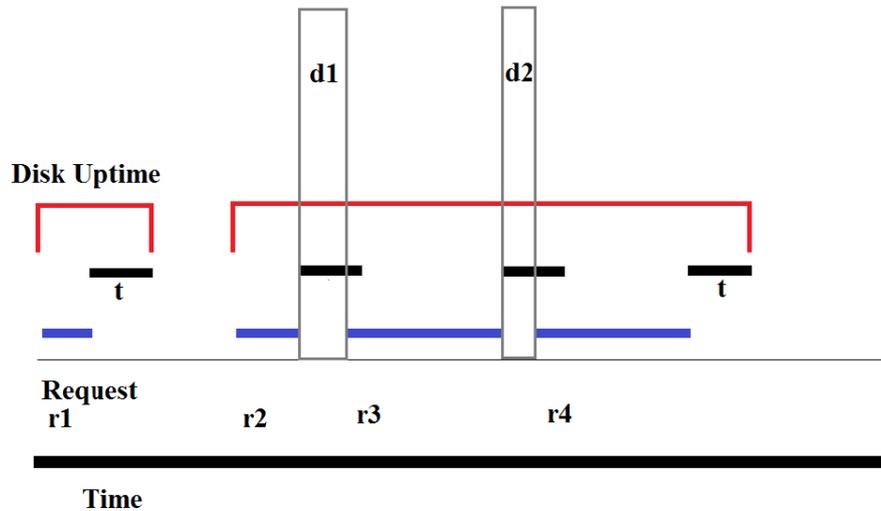


Fig. 2. Threshold based allocation method to Spin up of the Disk.

We consider a subset B which belongs to the set A , containing the requests which arrive in any given threshold time. The set B hence contains those “ m ” number of requests which combine with each other requests. Ex. r_2, r_3, r_4 in the example shown in Fig. 2. The total disk uptime is shown in red zones along the time line. The Differences show by d_1 and d_2 in the Fig. 2 represent the differences between the two consecutive overlapping requests under the given threshold time t_h . The difference of the m^{th} request from the set B can be shown by Eq. (4).

$$d_m = t_{a(m)} - t_{r(m-1)} - t_{a(m-1)} \quad (4)$$

Where $t_{a(n)}$ is the arrival time of the n^{th} request.

3.1. Runtime Cases

Under the worst case, the number of spin ups is equal to the number of requests, and the total Up Time can be represented by Eq. (5) as.

$$T_u = t_h * n + (\sum_{i=1}^n t_{r(i)}) \quad (5)$$

The average case DiskUp Time can be simplified further to represent the average of all d_m can be represented in terms of $t_h/2$ be shown in Eq. (6) as.

$$T_u = (n - m)t_h + m(t_h/2) + (\sum_{m \in B} t_{r(m)}) \quad (6)$$

The best case DiskUp Time can be shown in Eq. (7) as.

$$T_u = t_h + (\sum_{i=1}^n t_{r(i)}) \quad (7)$$

4. Simulation

The simulation is performed in Simulink of MATLAB and we use a DC Motor and a battery source to simulate the Hard Disk and the power consumption. We use an Ideal Switch which controls the circuit and feed the logic of the Ideal Switch via Signal Generator as shown in Fig. 3. The experimental data set taken in comparison of both methods is given in Table 1. We take the percentage SOC (State of Charge) of the battery in consideration and compare the two methods based on the amount of SOC Percentage left. The simulation runs for 1000 seconds and in a Discrete time unit of 1 second. The t_f from the data given in Table 1 is rounded off as 45 seconds. And t_h as 30 seconds. The Spin Up of the Hard Disk and the cylinder speed in Radians/Second which is a maximum of 750 which approximately equals 7161 RPM.

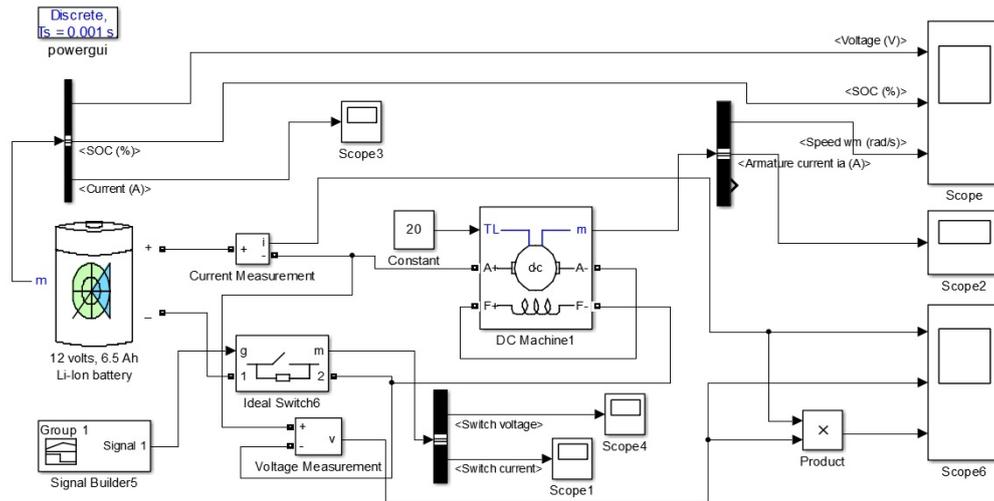


Fig 3. Simulation Model

Table 1. Data from a given Node A.

Request	Time of request	Arrival Time	Difference in case of Overlap	No. of Time frames allocated
1	80	56	-	2
2	95	403	-	3
3	102	518	20	2
4	60	711	-	2
5	112	810	-	3
Total	454			

4.1. Simulation of the Prediction Model

The timing values used for APRIORI Based Prediction model taken from Table 1, calculated via Eq. (2), Eq. (2) are fed to the control circuit using signal generator values as shown in Fig. 4. The Fig. 5 shows 3 graphs in APRIORI (Time Frame) based power management scheme, containing the Battery Voltage Drop, SOC decline with time. The Fig. 6 shows 3 graphs containing the Switch current, Switch voltage, Switch power time of the load considered to be the Hard Disk. The Final SOC after the simulation ends equals 45%.

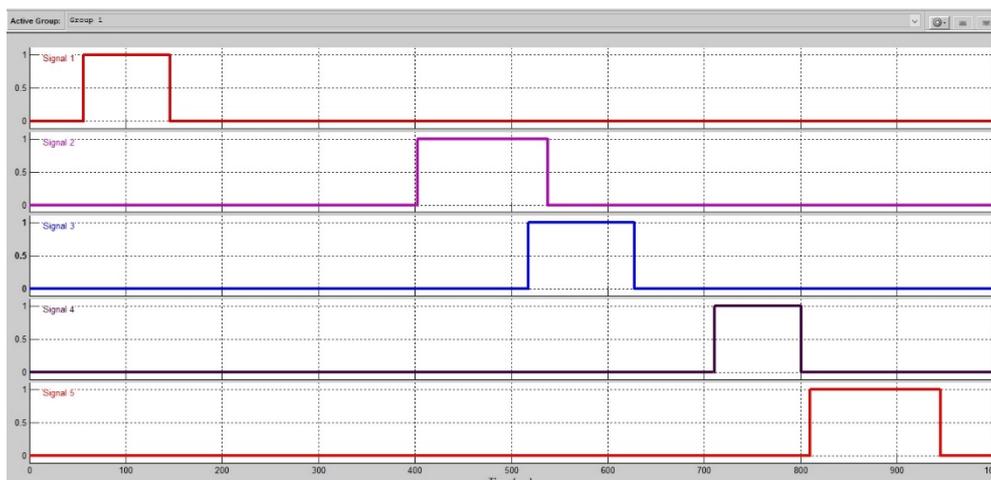


Fig. 4. Timing values of the Signal generator in Prediction Model

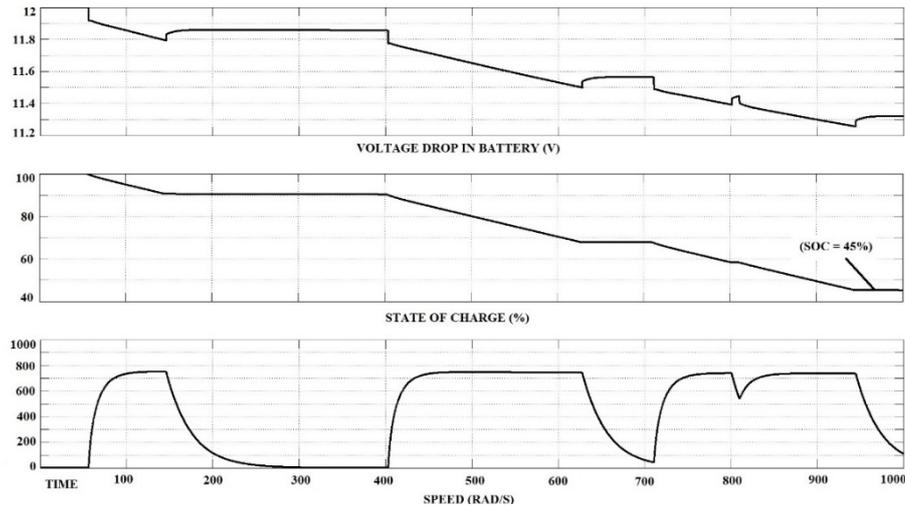


Fig. 5. Voltage Drop, SOC, Speed Switch in APRIORI, Time Frame based approach.

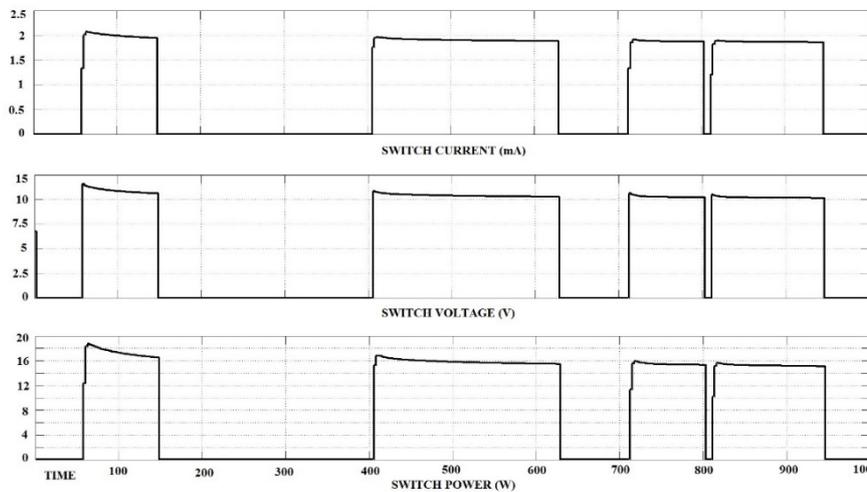


Fig. 6. Switch current, Switch voltage, Switch power in APRIORI, Time Frame based approach.

4.2. Simulation of the Threshold Model

The timing values used for Time Based Threshold model taken from Table 1, calculated via Eq. (4), Eq. (5), Eq (6) are fed to the control circuit using signal generator values as shown in Fig. 7. The Fig. 8 shows 3 graphs in Time threshold based power management scheme, containing the Battery Voltage Drop, SOC decline with time. The Final SOC after the simulation ends equals 40%.

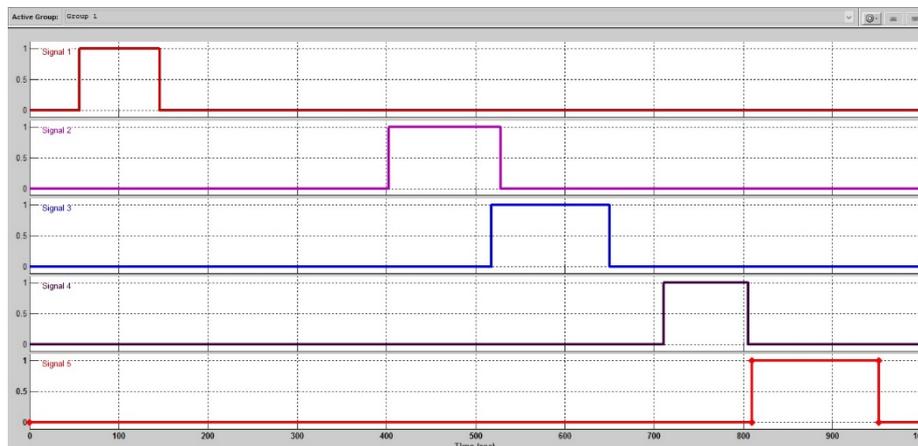


Fig. 7. Timing values of the Signal generator in Threshold Model

The Fig. 9 shows 3 graphs containing the Switch current, Switch voltage, Switch power time of the loads considered to be the Hard Disk.

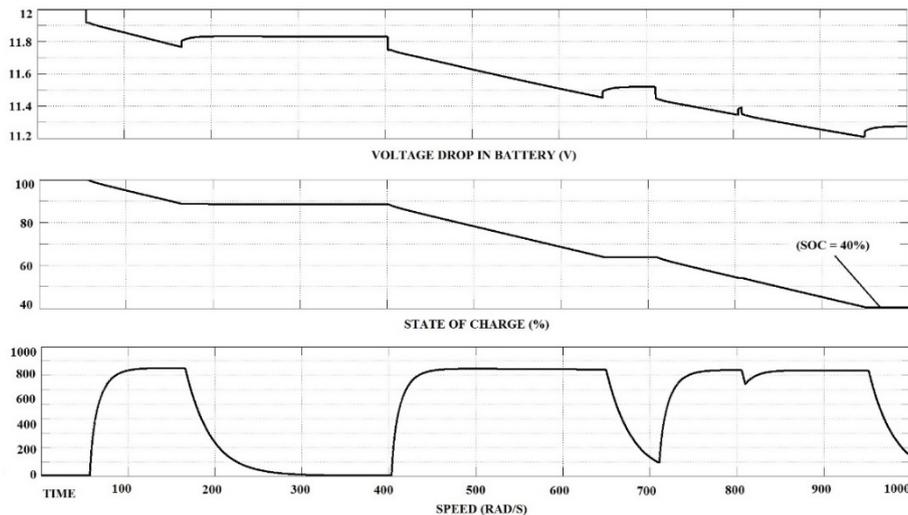


Fig. 8. Voltage Drop, SOC, Speed Switch in Time Threshold based approach.

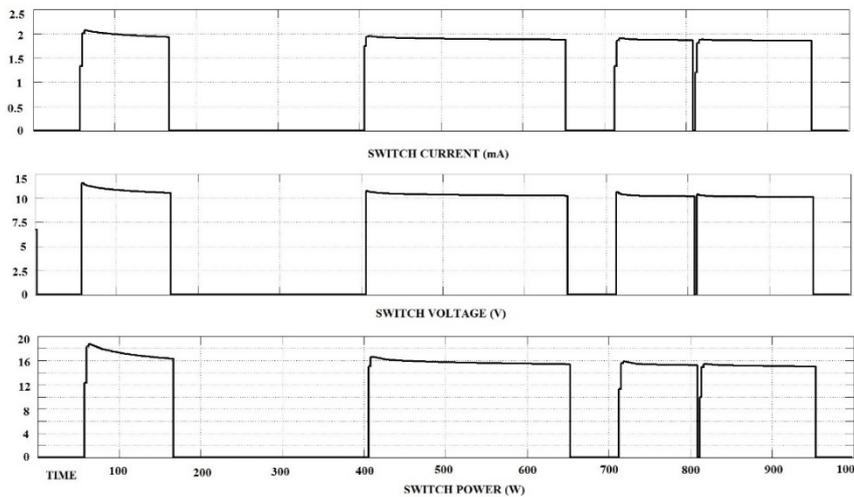


Fig. 9. Switch current, Switch voltage, Switch power in Time Threshold based approach.

Conclusion

In both the techniques discussed above, we can easily see that in APRIORI based management, the SOC is extra 5% than in the Time Threshold based approach. We can also conclude that even when both the approaches are almost equally efficient, the APRIORI based prediction method is less complicated in application. The constraints of the Data mining process are to be selected carefully as in power optimal scheduling and routing protocol[8], which tries to minimize in overall the total average power in the network. These values include Confidence and Support level and ideal selection of t_f for the cluster or a group of nodes, subjected to constraints such as peak transmission power of the nodes and achievable data rate per link.

Acknowledgments

The work done in the paper was under the guidance of Miss Amanpreet Kaur Assistant Professor, Yamuna Institute of Engineering and Technology, Haryana, India and Dr. Manpreet Professor HOD Electronics and Communication, also from the same institution, related to setup of simulation and editing.

References

- [1] F. Douglis, P. Krishnan, and B. Marsh: Thwarting the powerhungry disk. In *USENIX Winter Conference*, pages 293–306.
- [2] F. Douglis, P. Krishnan, and B. Bershada. Adaptive disk spindown policies for mobile computers. In *Computing Systems*, volume 8, pages 381–413, 1995.
- [3] Y. Ito and S. Kato, "An Apriori-Based Approach to Product Placement in Order Picking," 2016 IEEE International Conference on Agents (ICA), Matsue, Japan, 2016, pp. 114-115. doi: 10.1109/ICA.2016.039
- [4] M. Sanchez, P. Manzoni, and Z. J. Haas, "Determination of Critical Transmission Range in Ad Hoc Networks," *Proceedings of MMT 1999*, October 1999.
- [5] Yung-Hsiang Lu and G. De Micheli, "Adaptive hard disk power management on personal computers," *Proceedings Ninth Great Lakes Symposium on VLSI*, Ypsilanti, MI, 1999, pp. 50-53. doi: 10.1109/GLSV.1999.757375
- [6] C. F. Chiasserini and R. R. Rao, "A Distributed Power Management Policy for Wireless Ad Hoc Networks," *Proceedings of IEEE WCNC 2000*, vol. 3, pp. 1209-1213, September 2000.

- [7] S. Singh and C. S. Raghavendra, "Power-Aware Multi-Access Protocol with Signaling for Ad Hoc Networks," *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5-26, July 1998.
- [8] R. L. Cruz and A. R. Santhanam, "Optimal Routing, Link Scheduling, and Power Control in Multi-Hop Wireless Networks," *Proceedings of INFOCOM 2003*, vol. 1, pp. 702-711, April 2003.
- [9] Y. Chen, X. Du and L. Zhou, "Transformer defect correlation analysis based on Apriori algorithm," *2016 IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, Chengdu, 2016, pp. 1-4. doi: 10.1109/ICHVE.2016.7800686.
- [10] R. Zheng and R. Kravets, "On-Demand Power Management for Ad Hoc Networks," *Proceedings of IEEE INFOCOM 2003*, vol. 1, pp. 481-491, April 2003.