A Delay Sensitive Adaptive N-Policy Queueing System Design for Energy Efficient Sensor Network

Jie Chen
Hamad Bin Khalifa University
jchen@hbku.edu.qa

Biplab Sikdar
National University of Singapore
bsikdar@nus.edu.sg

Mounir Hamdi
Hamad Bin Khalifa University
hamdi@hbku.edu.qa

***Abstract*—A novel analytical model for N-policy queueing system is being proposed and tested against other established models and simulation results. Using the model, we argue that N-policy queueing system does not neccessarily save more energy as N increases.Based on the analytical and simulation results, optimal N selection scheme for a given input and service rate is being provided.Applying the above mentioned work to the sensor network, simulation results show that our work outperforms previous work in the area.Furthermore an adaptive N-policy system design is being illustrated and proves to save energy while not violating the delay constraint.**

I. Introduction

The portability of the end terminals in the current state of art network system have led to birth of the concept of green network. Due to the relatively short lifetime of battery that powers the end terminial, how to optimize the energy usage to facilitate a healthy and constructive mobile communication has been a critical issue in both academia and industry.Much of the work focus on duty cycle control [4], [5]. With the increase of the sleeping period, the average packet delay increases. It gives rise to another concern in the system design.

The communication system can not come out of blue. It is our decision not to ride the rising waves of the 5G standard popular in the wireless field but to refine well the analytical model in order to have fundamental breakthrough in the area of queueing system applicable to portable equipments. Henceforth, in our system, the portable equipments together with the aggregation point compose a basic sensor system. The packets are carried from the memory to the assembly line by proactively sensing packet queue workload.

Sensor system has its applications in many areas. It is widely used in home surveillance, supermarket self-check and other civil applications. Sensors, due to its limited power charge of battery, have posed constraints for its designs. Together with the expected customer experience of sensor system, power management scheme has become a major weight to measure the network performance against with. Power management schemes vary to different scenarios. For inter-device communication in the system, the transmission power is adjusted against the condition of the geographical channel. For intra-device communication in the system, the working power is switched between On and Off mode according to specific design configurations. Much work has been conducted for the latter case [4], [5], [7]–[9]. To determine how to switch between different power modes, the criteria has been set, for some as the duration of the working hours [4], [5], for some as the amount of workload that has been processed and for some as the amount of workload to be processed [**?**], [7]. With QOE the arrival of the upgraded QoS performance indicator in system design, N-policy queue as to process based on the workload to be done has become trendy.N-policy queue has been studied extensively in the past [**?**], [3]. How to find the optimal N to achieve a system design meeting the required statistical performance indicator retrieved from the general customer experience description has been a practical issue in the engineering field. To map the customer experience to the statistical performance indicator involves massive work in the field of psychology and computer design.Human computer interaction has become a much visited field recently. To determine the optimal N opens another door to analytical model design. These works evaluate the long term queue length distribution and use ***Little theorem*** to attain the average customer waiting duration in the queue and furthermore the average drop ratio. To be more specific, the long term expected queue length is decomposed into two parts : the working part when the server is on and the idle part when the server is off. Somehow, for the design sake, the system configuration is set once for all long. Intuitively, it is not scientifically feasible. Moreover, these work is theoretical and sometimes forgets to argue its applicability in real life. For example, paper [4] aims to design a power efficiency scheme for wireless sensor network but neglect in the scheme the finiteness of the buffer size per sensor. Furthermore, most of work takes for granted the traffic model from other source regarding the system design against expected system performance but forget the maxim that as far as the laws of mathematics refer to reality, they are not certain;and as far as they are certain, they do not refer to reality. More to argue the point is the fact that traffic modelling has itself been well-debated research area ever since.

Our research work aims to change the system configuration to adapt to the real-time traffic in order to save energy while not violating the delay constraint. In a simple design, the system configuration would be the buffer threshold. When the traffic is high, the system would resume to work more deligently, vice versa.

Our proposed analytical model evaluates the system performance in the long term. In the long term, the system statistics collected should be approximate to moments of the generating function. The system is being switched between on and off two

modes. It is our interest to evaluate the average time length of the system in both modes. From that point on, the system performance metrics could be derived and the threshold N could be adjusted iteratively.

In short, our work has four contributions:

- A new analytical model is being derived and tested well against the previous models in literature and also the simulation.
- Based on the analytical model, power consumption per threshold is being derived. Both analytical derivation and simulation prove that the power consumed does not decrease as the threshold N increases, given a fixed input rate.
- Comparisons of performance against the duty cycle control scheme is being conducted and shows positive results.
- A threshold N adjustment scheme is being proposed and shows to save more energy within a given delay bound.

To argue further, our work is practically theoretical and analytical workable.

II. Literature Review

Previous researches focus on the derivation of the system queueing analysis when the system is steady, given a well-established statistical traffic model. Literature has covered how to model the network traffic. It is not our intention here to investigate in the traffic modelling in details and further. Research on the duty cycle control for sensor system (T-policy queueing theory) has included the following work [4], [6].In paper [4] manage to achieve limited amount of backlog in the buffer by using the control principle while in paper [6]. Research that follows the case of N-policy queueing includes [**?**], [7]. Different cost function has been proposed. In paper [7], the cost composes of packet holding energy consumption while in paper [**?**], the cost includes transition power per cycle. It is worthwhile to note that in paper [7], if only considering the on mode energy consumption and off mode energy consumption, the power per threshold is a constant while in reality, as demonstrated well by simulation and mathematical analysis, the power per threshold varies and does not follow a deterministic pattern.Furthermore, to set the benchmark power as the one while $N = 1$ is not reasonable as $N = 1$ means the system would switch to sleeping mode when the buffer is empty. The correct benchmark would be the value when the system is always on - that is when $N = 0$. Research on N-policy queue theoretically could be catergorized as N-policy with setup time and N-policy without set up time. The former has its applicability to the real world situation when the server need time to warm up before the service while for the latter case, most of the work is devoted to generalize the policy into broader representations, such as to simulate the arrival as batch poisson or to simulate the service time as following a general probability distribution [3]. Paper [**?**]] has presented well the mathematical analysis of system performance metrics for different queueing policy.Paper [3] discussed further the mathematical common sense behind the complicated queueing length distribution incurred by adding the threshold. Another interesting branch that ascends from the N-policy queue is to optimize the system from the user's perspective that is to grant the user's strategic decision whether to enque or not based on the current state of the queue [10].Most of these work on N-policy is for server with finite capacity. Few work discusses the tradeoff between energy conservation and average packet delay.

III. System Model

Let the input flow follows poisson distribution with average rate λ and the scheduler departs the packet at a steady rate μ. The maximum queue length is K.The queue status is observed every $\frac{1}{\mu}$ interval. Suppose the sleeping queue wake-up threshold is N. Then

A. Model Jessica

For the queue length reaches threshold during the first observation interval

$$P_1 = 1 - \sum_{0}^{N-1} \exp -\frac{\lambda}{\mu} \cdot \frac{\frac{\lambda}{\mu}^k}{k!} \tag{1}$$

For the queue length reaches threshold during the second observation interval

$$P_2 = 1 - \sum_{0}^{N-1} \exp\left(-2 \cdot \frac{\lambda}{\mu}\right) \cdot \frac{\left(2 \cdot \frac{\lambda}{\mu}\right)^k}{k!} - P_1 \tag{2}$$

For the queue length reaches the nth observation interval

$$P_n = 1 - \sum_{0}^{N-1} \exp\left(-n \cdot \frac{\lambda}{\mu}\right) \cdot \frac{\left(n \cdot \frac{\lambda}{\mu}\right)^k}{k!} - \sum_{1}^{n-1} P_k \tag{3}$$

$$E(S_{idle}) = \sum_{1}^{\infty} i \cdot P_i \tag{4}$$

The distribution of queue length when the scheduler starts departing the packet

$$E(Q_{len}) = \sum_{N}^{K} i \cdot P(Q_{len} = i) = \sum_{N}^{K} i \cdot \sum_{j=1}^{\infty} P(Q_{len} = i, s = j) \tag{5}$$

$$\begin{aligned} &P(Q_{len} = i, s = j) \\ &= \sum_{k=0}^{N-1} P(k, \Delta t = (j-1) \cdot \frac{1}{\mu}) \cdot P((i-k), \Delta t = \frac{1}{\mu}) \\ &= \sum_{k=0}^{N-1} \exp -(j-1) \cdot \frac{\lambda}{\mu} \cdot \frac{\left[(j-1) \cdot \frac{\lambda}{\mu}\right]^k}{k!} \cdot \exp -\frac{\lambda}{\mu} \cdot \frac{\frac{\lambda}{\mu}^{(i-k)}}{(i-k)!} \\ &= \sum_{k=0}^{N-1} \frac{\exp\left(-j \cdot \frac{\lambda}{\mu}\right) \cdot \frac{\lambda}{\mu}^i \cdot (j-1)^k}{k! \cdot (i-k)!} \end{aligned} \tag{6}$$

The transition matrix is P, the dimesion of P is $K \times K$

$$p_{ij} = \begin{cases} 1, & \text{if } i == j == 0 \\ 0, & \text{if}(\ i == 0 \text{ and } j! = 0) \text{ or } \ j < (i-1) \\ \exp -\frac{\lambda}{\mu} \cdot \frac{\frac{\lambda}{\mu}^{j-i+1}}{(j-i+1)!}, & \text{if } (i-1) \leq j < K \\ 1 - \sum_{k=0}^{K-1} p_{ik}, & \text{otherwise} \end{cases} \tag{7}$$

For the scheduler to work, the initial distribution of queue length is P_{init}, the dimension of which is $1 \times K$.

$$P_{init}(i) = \sum_{j=1}^{\infty} P(Q_{len} = i, s = j) \tag{8}$$

For the scheduler to stop while the queue becomes empty during $1st$ observation interval,

$$P_1(0) = P_{init}(0) \tag{9}$$

$$P_i = P'_{(i-1)} \cdot P \tag{10}$$

$$P'_{(i-1)}(k) = \begin{cases} 0, & \text{if } k == 0 \\ P_{(i-1)}(k), & \text{otherwise} \end{cases} \tag{11}$$

$$E(S_{active}) = \sum_{i=0}^{\infty} i \cdot P_i(0) \tag{12}$$

Let $\mathbf{E(Arrival_N)}$ be the average total amount of packets that arrive during one cycle, then

$$\begin{aligned} &\mathbf{E(Arrival_N)} \\ &= E[Arrival_{N_{idle}} + Arrival_{N_{active}}] \\ &= E[Arrival_{N_{idle}}] + E[Arrival_{N_{active}}] \\ &= \mathbf{E(Q_{len})} + \sum_{i=0}^{\infty} i \cdot P(Arrival = i) \\ &= \mathbf{E(Q_{len})} + \sum_{i=0}^{\infty} \sum_{n=1}^{\infty} P(Arrival = i, t_{active} = \frac{n}{\mu}) \\ &= \mathbf{E(Q_{len})} + \\ &\sum_{i=0}^{\infty} i \cdot \sum_{n=1}^{\infty} P(Arrival = i | t_{active} = \frac{n}{\mu}) \cdot P(t_{active} = \frac{n}{\mu}) \\ &= \mathbf{E(Q_{len})} + \\ &\sum_{n=1}^{\infty} [\sum_{i=0}^{\infty} i \cdot P(Arrival = i | t_{active} = \frac{n}{\mu})] \cdot P(t_{active} = \frac{n}{\mu}) \\ &= \mathbf{E(Q_{len})} + \sum_{n=1}^{\infty} \lambda \cdot \frac{n}{\mu} \cdot P(t_{active} = \frac{n}{\mu}) \\ &= \mathbf{E(Q_{len})} + \frac{\lambda}{\mu} \cdot \mathbf{E(S_{active})} \end{aligned} \tag{13}$$

Let $\mathbf{E(Depatl_N)}$ be the average amount of packets that depart during one cycle, then

$$\begin{aligned} &\mathbf{E(Depart_N)} \\ &= \sum_{j=N}^{\infty} j \cdot P(t_{active} \cdot \mu = j) \\ &= \sum_{j=N}^{\infty} P(t_{active} = \frac{j}{\mu}) \cdot \frac{j}{\mu} \cdot \mu \\ &= \sum_{j=N}^{\infty} P(s_{active} = j) \cdot \frac{j}{\mu} \cdot \mu \\ &= \mathbf{E(s_{active})} \end{aligned} \tag{14}$$

Let $\mathbf{E(drop)}$ be the expectation of the drop ratio for a running cycle of queue length from 0 to 0, then

$$\begin{aligned} &\mathbf{E(drop)} \\ &= \frac{E(Arrival_N) - E(Departure_N)}{E(Arrival_N)} \\ &= 1 - \frac{E(Departure_N)}{E(Arrival_N)} \\ &= 1 - \frac{\mathbf{E(s_{active})}}{\mathbf{E(Q_{len})} + \frac{\lambda}{\mu} \cdot \mathbf{E(s_{active})}} \end{aligned} \tag{15}$$

Let $\mathbf{E(latency)}$ be the expectation of the latency for a running cycle of queue length from 0 to 0, then

$$\begin{aligned} &\mathbf{E(latency)} \\ &= \frac{\sum_1^{E(Depart_N)} (E[Departure_i] - E[Arrival_i])}{E(Depart_N)} \\ &= \frac{\sum_1^{E(Depart_N)} (E[t_{idle} + \frac{i}{\mu}] - E[\sum_{k=1}^{i} interarrivaltime_k])}{E(Depart_N)} \\ &= \frac{\sum_1^{E(Depart_N)} (E[t_{idle}] + \frac{i}{\mu} - \frac{i}{\lambda})}{E(Depart_N)} \\ &= \frac{E[t_{idle}] \cdot E(Depart_N) + (\frac{1}{\mu} - \frac{1}{\lambda}) \cdot \frac{[E(Depart_N)+1] \cdot E(Depart_N)}{2}}{E(Depart_N)} \\ &= \mathbf{E(t_{idle})} + (\frac{1}{\mu} - \frac{1}{\lambda}) \cdot \frac{E(Depart_N) + 1}{2} \\ &= \frac{\mathbf{E(s_{idle})}}{\mu} + (\frac{1}{\mu} - \frac{1}{\lambda}) \cdot \frac{\mathbf{E(s_{active})} + 1}{2} \end{aligned} \tag{16}$$

It is also intuitively simple to reach the derivation that $E(s_{idle}) \approx \mu \cdot \frac{N}{\lambda}$. Suppose t is the active period when the queue length changes from $\sim N$ to 0. Then,the following equation makes sense.

$$\begin{aligned} &t \cdot \lambda + N = t \cdot \mu \\ &t = \frac{N}{\mu - \lambda} \end{aligned} \tag{17}$$

As $t \cdot \mu = E(s_{active})$, then

$$\begin{aligned}&\mathbf{E(latency)}\\&= \frac{N}{\lambda} + (\frac{1}{\mu} - \frac{1}{\lambda}) \cdot \frac{1 + \frac{N}{\mu - \lambda} \cdot \mu}{2}\\&= \frac{1}{2 \cdot \mu} + \frac{N-1}{2 \cdot \lambda}\end{aligned} \tag{18}$$

Thus so far two formulas for average packet latency are attained and according to Fig. IV-A and Fig. IV-A, the difference between the analytical results attained from the formulas are insignificant and both are approximately the same as the results attained from simulation.The other formula we used to compare is borrowed from paper [**?**]. It is tagged as Jayaparvathy model.

$$\begin{aligned}&\mathbf{E(Energy)}\\&= \frac{Power_{on} \cdot E(t_{active}) + Power_{off} \cdot E(t_{idle})}{E(t_{active}) + E(t_{idle})}\\&= \frac{Power_{on} \cdot \frac{N}{\mu - \lambda} + Power_{off} \cdot \frac{N}{\lambda}}{\frac{N}{\mu - \lambda} + \frac{N}{\lambda}}\\&= Power_{on} \cdot (1 - \frac{\lambda}{\mu}) + Power_{off} \cdot \frac{\lambda}{\mu}\end{aligned} \tag{19}$$

A conclusion could be reached that the average power consumed is constant regardless of the value of N. Firstly, it disobeys our intention of our design. Secondly from Fig. IV-A and Fig. IV-A, discernable variations of the power value against different N exist. Henceforth, the average power per threshold is being discussed necessarily as below:

Suppose the observation timespan is t, the average busy steps per cycle is BC and the average idle steps per cycle is IC, then on average there would be m cycles, in which

$$m = \frac{t \cdot \mu}{BC + IC} \tag{20}$$

$m = k + \beta$, $k = \lfloor m \rfloor$. Then $E = E_1 + E_2$.

$$E_1 = k \cdot [Power_{on} \cdot BC + Power_{off} \cdot IC] \tag{21}$$

$$E_2 = \begin{cases} Power_{off} \cdot \beta \cdot (BC + IC), \\ \text{if } \beta \cdot (BC + IC) \leq IC \\ Power_{off} \cdot IC + [\beta \cdot (BC + IC) - IC] \cdot Power_{on}, \\ \text{otherwise} \end{cases} \tag{22}$$

$$Power = \begin{cases} \frac{k \cdot (Power_{on} \cdot BC + Power_{off} \cdot IC)}{(k+\beta) \cdot (BC + IC)} + & \text{if } \beta \leq \alpha \\ \frac{Power_{off} \cdot \beta \cdot (BC+IC)}{(k+\beta) \cdot (BC+IC)}, & \\ \frac{k}{k+\beta} \cdot \frac{Power_{on} \cdot BC + Power_{off} \cdot IC}{BC + IC} + & \\ \frac{[\beta \cdot (BC+IC) - IC] \cdot Power_{on}}{(k+\beta) \cdot (BC+IC)} + & \text{otherwise} \\ \frac{Power_{off} \cdot IC}{(k+\beta) \cdot (BC+IC)} & \end{cases} \tag{23}$$

$\alpha = \frac{IC}{BC+IC}$

Further proofs are being presented in the appendix.

B. Model Takagi

A conventional method to approach the above problem is to observe the system at time instance when either the server ends its idle period or the server completes a service.Average queue length at an arbitrary time instance is attained in order to caculate the average delay and also the drop ratio. This model borrowed mathematical notations from paper [**?**]. Details are illustrated as follows.

Let ξ_n be the nth observation instance and

$$\xi_n = \begin{cases} 0, & \text{a idle period ends} \\ 1, & \text{a service completes} \end{cases} \tag{24}$$

Let L_n be the queue length at nth time instance

$$q_k = \lim_{n \to \infty} Prob[\xi_n = 0, L_n = k], 0 \leq k \leq K \tag{25}$$

$$\pi_k = \lim_{n \to \infty} Prob[\xi_n = 1, L_n = k], 0 \leq k \leq K \tag{26}$$

$$q_k = \pi_0 \cdot P_{init}(k), 0 \leq k \leq K \tag{27}$$

$$\pi_k = \sum_{i=1}^{k+1} (\pi_i + q_i) \cdot a_{i,k}, 0 \leq k \leq K \tag{28}$$

$$a_{i,k} = \begin{cases} \exp(\frac{-\lambda}{\mu}) \cdot \frac{\frac{\lambda}{\mu}^{k+1-i}}{(k+1-i)!}, & (i \leq k-1) \text{and } (k \neq K) \\ 0, & (i < k-1) \text{and } (k \neq K) \\ 1 - \sum_{n=0}^{K-1} a_{i,k} & \text{otherwise} \end{cases} \tag{29}$$

Let $P = [\pi_0, \pi_1, \ldots, \pi_K]$, $P' = [q_0, q_1, \ldots, q_K]$ and A is a $\mathbf{K} \times \mathbf{K}$ matrix with $A_{i,j} = a_{i,k}$,$0 \leq i, j = k \leq \mathbf{K - 1}$

$$\begin{aligned}&P' = \pi_0 * P_{init}\\&= P * X * P_{init}\\&X_{i,j} = \begin{cases} 1, & \text{if} i == j == 0 \\ 0 & \text{otherwise} \end{cases}\end{aligned} \tag{30}$$

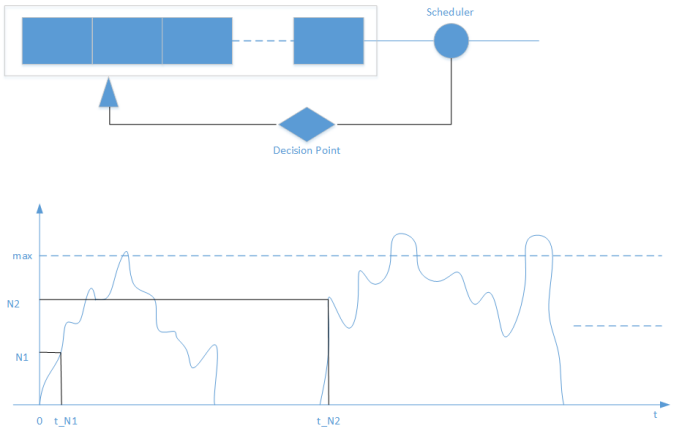


Fig. 1. Adaptive N policy scheduler

$$A' = T * A$$
$$T_{i,j} = \begin{cases} 0, & \text{if } i! = j \text{or} 1 == j == 0 \\ 1 & \text{otherwise} \end{cases} \tag{31}$$

$$\begin{aligned} &(P + P') * A' = P \\ &(P + P * X * P_{init}) * A" = P \\ &P * (A' + X * P_{init} * A') = P \\ &P * (T * A + X * P_{init} * T * A) = P \end{aligned} \tag{32}$$

Let $B = T * A + X * P_{init} * T * A$, then

$$P * B = P \tag{33}$$

According to paper [**?**],

$$\mathbf{E(drop)} = 1 - \frac{1 - \pi_0}{\frac{\pi_0 \cdot N \cdot \mu}{\lambda} + (1 - \pi_0)} \tag{34}$$

$$\begin{aligned} \mathbf{E(latency)} = &\frac{\sum_{k=1}^{K-1} k \cdot \pi_k}{(1 - \pi_0) \cdot \lambda} + \\ &\frac{K}{\lambda} \cdot (\frac{\pi_0 \cdot N + (1 - \pi_0) \cdot \frac{\lambda}{\mu}}{1 - \pi_0} - 1) - \frac{1}{\mu} \end{aligned} \tag{35}$$

From Fig. IV-A and Fig. IV-A, Takagi model performs badly when the input rate is low and acts approximately the same as the rest models when the input rate increases.

For the rest of the paper, model Jessica and its simplified version would be used.

C. Adaptive Optimal Threhold Selection

Algorithm 1.

TABLE IV-A and TABLE IV-A shows that analytical results for low input rate are the same or very close to the simulation results. As N-policy works only for system that is unsaturated. The algorithm is henceforth applicable to reality.

Once algorithm is validated, it is our interest to change it adaptively to the real traffic stream, in which case, the average

Algorithm 1: Optimal N-search Algorithm

Input: D:Expected Latency
Output: N: Optimal N

$temp_p \leftarrow P_{on}$,$opt_n \leftarrow 0$,$t \leftarrow T$
$N \leftarrow \lfloor (D - \frac{1}{2 \cdot \mu}) \cdot 2 \cdot \lambda + 1 \rfloor$ **foreach** n *in* N **do**
  if $Power_n \leq temp_p$ **then**
    $temp_p = Power_n$
    $opt_n = n$
return opt_n

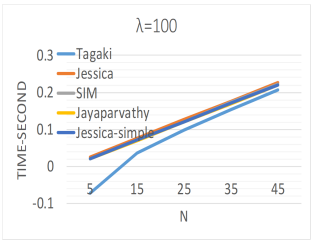


Fig. 2. lambda=100 - Delay

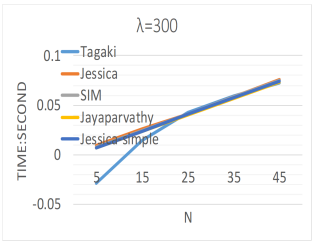


Fig. 3. lambda=300 - Delay

rate for a given period varies from the long term value. When there exists high volume of data coming to the service queue, it is expected that the server could wake from the sleeping period sooner. Intuitively, the threshold N would become smaller. As shown in Fig. 1, the packet scheduler changes the threshold value based on the packet input rate.The scheduler periodically monitor the average flow rate. Once there is a change, the scheduler would attain the optimal N based on the current flow information at hand(a straightforward metrics are the flow rate and flow timespan).The comparisons are conducted between the power consumptions within a fixed delay bound of adaptive scheme and the origin scheme. For the origin scheme, thresholds are picked as those optimal for individual flow of packets. Results are shown in Fig. IV-A. It is noticeable that for the given delay upper bound *0.2 s*, the adaptive scheme works better than the original plain scheme.

IV. Simulation Results

A. Byun vs Jessica

The simulation configuration is according to the one-hop network topology in paper [4].The comparisons of both delay and power indicate that Model Jessica outperforms the model proposed by Dr.Byun and Dr.Yu in paper [4].Fig. 6 and Fig. 7 show the point clearly.

TABLE I
Optimal Threshold-lambda=10

D	0.5	1	1.5	2
ANA	8	16	27	32
SIM	7	17	25	39

TABLE II
Optimal Threshold-lamba=50

D	0.1	0.2	0.3	0.4
ANA	9	18	27	36
SIM	7	17	29	37

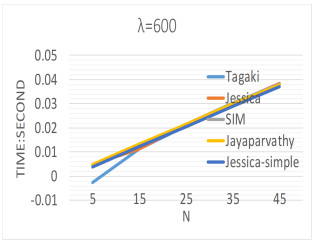


Fig. 4. lambda=600 - Delay

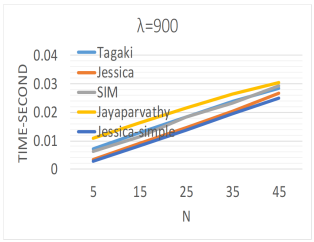


Fig. 5. lambda=900 - Delay

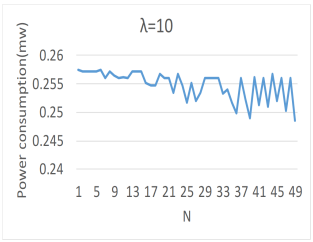


Fig. 8. lambda=10 - Power

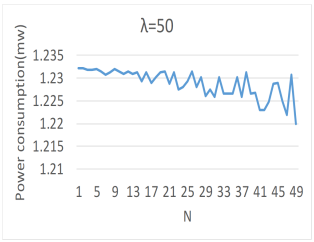


Fig. 9. lambda=50 - Power

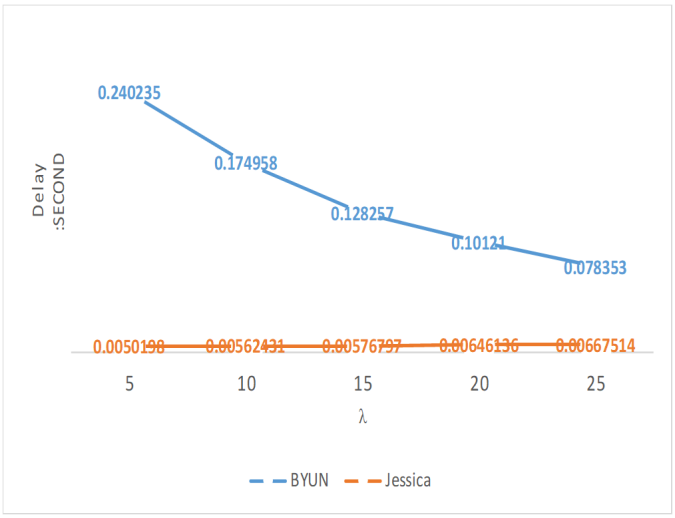


Fig. 6. Cycle Control vs N-policy - Delay

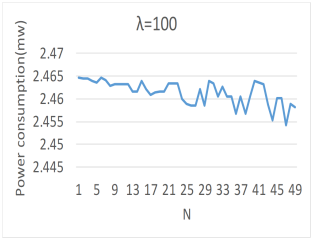


Fig. 10. lambda=100 - Power

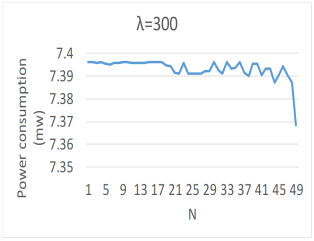


Fig. 11. lambda=300 - Power

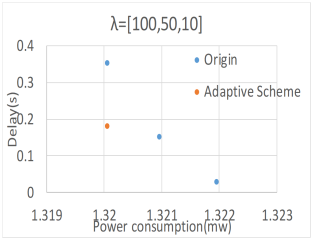


Fig. 12. Delay Upper Bound=0.2

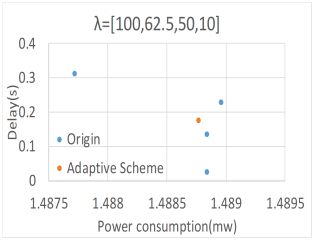


Fig. 13. Delay Upper Bound=0.2

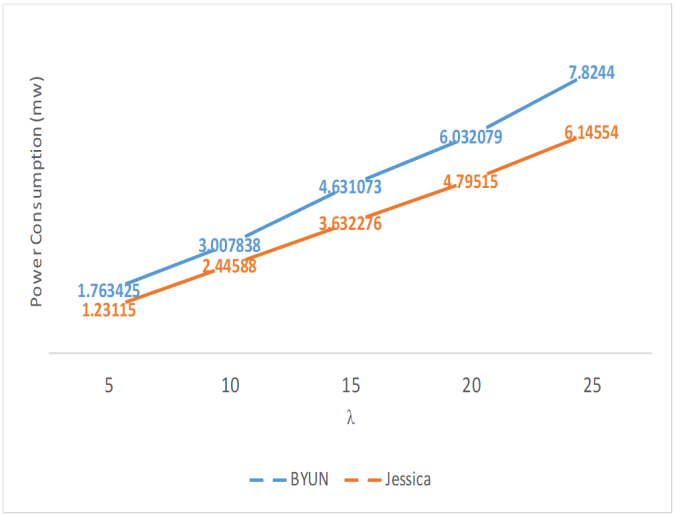


Fig. 7. Cycle Control vs N-policy - Power

TABLE III
OPTIMAL
THRESHOLD-LAMBDA=100

D	0.1	0.2	0.05	0.01
ANA	19	38	7	1
SIM	17	38	8	2

TABLE IV
OPTIMAL THRESHOLD-LAMBA=600

D	0.01	0.02	0.03	0.035
ANA	7	23	33	42
SIM	11	18	35	38

APPENDIX

Here we give the proof that N-policy would not guarantee a save on energy - evaluated within a fixed timeframe. Assume a fixed average input rate and departure rate, when the threshold is increased from N_1 to N_2. It results into two different pairs of (k_i, β_i).

If $\beta_1 \leq \alpha$ and $\beta_2 \leq \alpha$ then,

$$P_{N_1} - P_{N_2} = \frac{P \cdot k_1 + P_off \cdot \beta_1}{k_1 + \beta_1} - \frac{P \cdot k_2 + P_off \cdot \beta_2}{k_2 + \beta_2} = \frac{k_1 \cdot \beta_2 - k_2 \cdot \beta_1}{(k_1 + \beta_1) \cdot (k_2 + \beta_2)} \cdot (P - P_off) \tag{36}$$

case 1 : let $(k_1, \beta_1) = (5, 0.2)$ and $(k_2, \beta_2) = (3, 0.5)$, if $t = 10$,$\mu = 100$ and $\lambda = 10$, then $N_1 = 17$ and $N_2 = 26$. $N_1 > N_2$. The precondition exists. $P_{N_1} > P_{N_2}$

case 2: let $(k_1, \beta_1) = (5, 0.2)$ and $(k_2, \beta_2) = (3, 0.1)$, if $t = 10$,$\mu = 100$ and $\lambda = 10$, then $N_1 = 17$ and $N_2 = 29$. $N_1 \leq N_2$. The precondition exists.$P_{N_1} \leq P_{N_2}$

If $\beta_1 > \alpha$ and $\beta_2 > \alpha$ then,

$$P_{N_1} - P_{N_2} = \frac{(P_{on} - P) \cdot (\beta_1 \cdot k_2 - \beta_2 \cdot k_1) + \alpha * (P_{off} - P_{on})(k_2 + \beta_2 - k_1 - \beta_1)}{(k_1 + \beta_1) \cdot (k_2 + \beta_2)}$$
$$= \frac{\alpha \cdot (P_{on} - P_{off}) \cdot (\beta_1 \cdot k_2 - \beta_2 \cdot k_1 - k_2 - \beta_2 + k_1 + \beta_1)}{(k_1 + \beta_1) \cdot (k_2 + \beta_2)} \tag{37}$$

Apply case1 and case2 above, the same conclusion is reached that P_{N-1} could be either bigger or smaller than P_N.

If $\beta_1 \leq \alpha$ and $\beta_2 > \alpha$ then,

$$P_{N_1} - P_{N_2} = \frac{P \cdot k_1 + P_{on} \cdot \beta_1}{k_1 + \beta_1} - \frac{P \cdot k_2 + P_{on} \cdot \beta_2 + \alpha \cdot (P_{off} - P_{on})}{k_2 + \beta_2}$$
$$= \frac{k_1 \cdot \beta_2 \cdot (P - P_{on}) + \beta_1 \cdot k_2 \cdot (P_{off} - P) + \beta_1 \cdot \beta_2 \cdot (P_{off} - P_{on}) - (k_1 + \beta_1) \cdot \alpha \cdot (P_{off} - P_{on})}{(k_1 + \beta_1) \cdot (k_2 + \beta_2)}$$
$$= (P_{on} - P_{off}) \frac{-k_1 \cdot \beta_2 \cdot \alpha - \beta_1 \cdot k_2 \cdot (1 - \alpha) + [\alpha \cdot (k_1 + \beta_1) - \beta_1 \cdot \beta_2]}{(k_1 + \beta_1) \cdot (k_2 + \beta_2)} \tag{38}$$

Suppose$\alpha = 0.6$,$(k_1, \beta_1) = (5, 0.1)$ and $(k_2, \beta_2) = (1, 0.7)$, apply the values to the above equation, $P_{N_1} < P_{N_2}$; however with $(k_1, \beta_1) = (8, 0.1)$ and $(k_2, \beta_2) = (1, 0.7)$,$P_{N_1} > P_{N_2}$. In both cases, the precondition $N_1 \leq N_2$ exist.

The same apply to the if branch $\beta_1 > \alpha$ and $\beta_2 \leq \alpha$.

REFERENCES

[1] J.R.Norris, "Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics)" *Cambridge University Press*1997

[2] H.Takagi, "M/G/1/K queues with N-policy and setup times"*Queueing Systems* 1993

[3] H.Lee, B.Ahn and N.Park,"Decompositions of the Queue Length Distributions in the MAP/G/1 Queue under Multiple and Single Vacations with N-policy"*Stochastic Models vol.17 iss.2*2001

[4] H. Byun and J.Yu, "Adaptive Duty Cycle Control with Queue Management in Wireless Sensor Networks"*IEEE TRANSACTIONS ON MOBILE COMPUTING vol.12, no.6* June 2013

[5] Y.Li,K.Chai,Y.Chen and J.Loo, "Duty cycle control with joint optimization of delay and energy efficiency for capillary machine to machine networks for 5G communication system" *Transactions on Emerging Telecommunications Technologies*2015

[6] Z.Chen,A.Liu,Z.Li,Y.Choi and J.Li, "Distributed duty cycle control for delay improvement in wireless sensor networks"*Peer-to-Peer Networking and Applications vol.10, iss.3* May 2017

[7] F.Jiang, C.Yang and F.Lei, "Lifetime elongation for wireless sensor network using queue-based approaches" *The Journal of SuperComputing* March 2012

[8] R.Maheswar and R.Jayaparvathy,"Power Control Algorithm for Wireless Sensor Networks using N-Policy M/M/1 Queueing Model" *International Journal on Computer Science and Engineering vol.2, no.7* 2010

[9] W.M. Kempa, "On Queueing Delay in WSN with Energy Saving Mechanism Based on Queued Wake Uo"*IWSSIP 2014, 21st International Conference on Systems, Signals and Image Processing* May 2014

[10] J.Wang,X.Zhang and P.Huang, "Strategic behavior and social optimization in a constant retrial queue with the N-policy"*European Journal of Operation Research vol.256, iss.3* Feb. 2017