

M/G/1 Vacation Queueing Systems with Server Timeout

Oliver C. Ibe

Department of Electrical and Computer Engineering, University of Massachusetts, Lowell, USA
Email: oliver_ibe@uml.edu

Received 21 February 2015; accepted 11 March 2015; published 13 March 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

We consider a single-server vacation queueing system that operates in the following manner. When the server returns from a vacation, it observes the following rule. If there is at least one customer in the system, the server commences service and serves exhaustively before taking another vacation. If the server finds the system empty, it waits a fixed time c . At the expiration of this time, the server commences another vacation if no customer has arrived; otherwise, it serves exhaustively before commencing another vacation. Analytical results are derived for the mean waiting time in the system. The timeout scheme is shown to be a generalized scheme of which both the single vacation and multiple vacations schemes are special cases, with $c = \infty$ and $c = 0$, respectively. The model is extended to the N -policy vacation queueing system.

Keywords

Vacation Queueing Systems, Timeout Policies, Performance Analysis, N -Policy with Timeout

1. Introduction

A vacation queueing system is one in which a server may become unavailable for a random period of time from a primary service center. The time away from the primary service center is called a vacation, and customers who arrive while the server is on vacation will have to wait until he returns from vacation. A vacation can be the result of many factors. In some cases, the vacation can be the result of server breakdown, which means that the system must be repaired and brought back to service. It can also be a deliberate action taken to utilize the server in a secondary service center when there are no customers present at the primary service center. Thus, server vacations are useful for those systems in which the server's idle time is utilized for other purposes, and this makes the queueing model to be applicable to a variety of real world stochastic service systems.

One example of the application of vacation queueing systems is in token-passing local networks. In this case,

a station can only transit its packets when it receives the token, which essentially corresponds to the server returning from a vacation. After transmitting the prescribed number of packets, the station passes the token to the next station in the transmitting order; this action is equivalent to the server taking a vacation with respect to that station. The station will transmit again when next it receives the token; or equivalently, when the server returns from vacation. In general, any system where service is provided in a round robin manner can be modeled by a vacation queueing system. Other applications that can be modeled by the vacation queueing system include machine breakdowns and maintenance in communication and computer systems. In these two cases, the server is unavailable to serve customers when the system breaks down or is intentionally taken out of service for maintenance. Thus, relative to the customers the server is on vacation.

There are different types of vacation queueing systems. In the *single vacation scheme*, the server takes a vacation of a random duration when the queue is empty. At the end of the vacation, the server returns to the queue. If there is at least one customer waiting when the server returns from vacation, the server performs one of the following actions depending on the service policy:

a) Under the *exhaustive service* policy, the server will serve all waiting customers as well as those that arrive while he is still serving at the station. He takes another vacation when the queue becomes empty.

b) Under the *gated service* policy, the server will serve only those customers that he finds at the queue upon his return from vacation. At the end of their service, the server will commence another vacation and any customers that arrive while the server is already serving at the station will be served when the server returns from the vacation.

c) Under the *limited service* policy, the server will serve only a predefined maximum number of customers and then will commence another vacation. The single-service scheme in which exactly one customer is served is a special type of this policy.

If the queue is empty on the server's return, the server waits to complete a busy period using one of the above service policies before taking another vacation.

In the *multiple vacation scheme*, if the server returns from a vacation and finds the queue empty, he immediately commences another vacation. If there is at least one waiting customer, then he will commence service according to the prevailing service policy.

Queueing systems with server vacations have attracted the attention of many researchers since the idea was first discussed in the paper of Levy and Yechiali [1]. Several excellent surveys on these vacation models have been done by Doshi [2] [3], and the books by Takagi [4] and Tian and Zhang [5] are devoted to the subject. One of the most remarkable results for vacation queueing systems is the stochastic decomposition result, which is introduced by Fuhrmann and Cooper [6] and allows the system's behavior to be analyzed by considering separately the distribution of system (queue) size with no vacation and additional system size due to vacation.

Many new vacation queueing systems have been proposed. In the vacation queueing system, we have described the server completely stops service or is switched off when he is on vacation. Recently, Servi and Finn [7] introduced the *working vacation scheme*, in which the server worked at a different rate rather than completely stopping service during a vacation. The working vacation scheme has attracted a lot of research effort, and several authors have extended the original model. Wu and Takagi [8] generalized the model in [7] to an M/G/1 queue with general working vacations. Baba [9] studied a GI/M/1 queue with working vacations by using the matrix analytic method. Banik, Gupta and Pathak [10] analyzed the GI/M/1/N queue with working vacations. Liu, Xu and Tian [11] established a stochastic decomposition result in the M/M/1 queue with working vacations. In [12], Krishnamoorthy and Sreenivasan considered an M/M/2 queueing system with heterogeneous servers where one server is always available and the other goes on a working vacation when there are no customers in the system.

The *differentiated multiple vacation queueing* system is another vacation queueing model that is proposed by Ibe and Isijola [13]. The authors considered an M/M/1 multiple vacation queueing system with two types of vacations: Type 1 vacation is taken when the server returns from a vacation and finds at least one waiting customer; it takes the vacation after exhaustively serving all customers in the system. Type 2 vacation is taken when the server returns from a vacation and finds no customers in waiting to be served. The duration of one type of vacation is different from that of the other type.

Another model of the vacation queueing system is the system with *vacation interruption* in which the server can come back to the normal working level before the vacation ends. Li and Tian [14] analyzed an M/M/1 queue with working vacations and vacation interruption. The authors also extended the model to the GI/Geo/1 queue with working vacation and vacation interruptions [15]. The GI/M/1 queue with working vacations and vacation in-

terruption was studied by Li, Tian and Ma [16]. Similarly, Zhang and Hou [17] discussed an M/G/1 queue with multiple working vacations and vacation interruption. In [18], Sreenivasan, Chakravarthy and Krishnamoorthy considered a single server queueing model in which customers arrived according to a Markovian arrival process. They also introduces a threshold, N , where $1 \leq N < \infty$, such that the server offering services (at a lower rate) during a vacation will have the vacation interrupted the moment the queue size reaches N . In [19] Isijola-Adakeja and Ibe extended the model on [13] to include server vacation interruption.

In the vacation models that have been analyzed in the literature, server timeouts have not been considered. In this paper, we consider vacation queueing systems with server timeouts. Specifically, we consider a system that operates in the following manner. When the server has finished serving a customer and finds the system empty, it takes a vacation whose duration is independent of both the service time and the inter-arrival time. At the end of the vacation, the server returns to serve those customers, if any, who arrive during its vacation. It will commence another vacation when the system becomes empty. If no customer arrives during the vacation, the server waits for a fixed time c . If no customer arrives by the end of this period, the server commences another vacation. If a customer arrives before the period expires, the server commences service and serves exhaustively before commencing another vacation.

The paper is organized as follows. The model is formally defined in Section 2 and analyzed in Section 3. An extension of the model to the N -policy scheme is given in Section 4, and concluding remarks are made in Section 5.

2. Model Description

Figure 1 illustrates this scheme. Assume that the system has been operational for some time and the server returns from a vacation of duration V_1 at a time labeled t_1 and found two customers C_1 and C_2 waiting. Before completing the service of these customers, another customer C_3 arrives. After serving the three customers the server takes another vacation of duration V_2 . Upon returning from that vacation at a time t_2 it found no customer waiting. It waits for a time of duration c without any customer arriving. It thus leaves for another vacation of duration V_3 without serving a customer. It came back from the vacation at the time labeled t_3 and found two customers C_4 and C_5 waiting. Before completing the service of these customers, another customer C_6 arrives. After serving these three customers, the server takes another vacation of duration V_4 . Upon returning from the vacation at the time labeled t_4 , it found no waiting customers. However, before the expiration of the timeout, customer C_7 arrives to initiate another busy period, and the process continues.

This vacation queueing is essentially a hybrid multiple and single vacation scheme that was introduced in Ibe and Trivedi [20]. The main objective of the model in [20] was to demonstrate how vacation queueing systems with exponentially distributed service times and finite population could be modeled by the stochastic Petri net. In this paper we assume that the population is infinite and service times have arbitrary distribution.

One important application of the vacation scheme with server timeout is to enhance resource utilization in the single vacation model. Specifically, if there is a problem in the arrival process that prevents customers from arriving for service, the server may be idle indefinitely after returning from a vacation in the single vacation. But when the idle time is bounded as described above, the server can be used to perform other functions at the

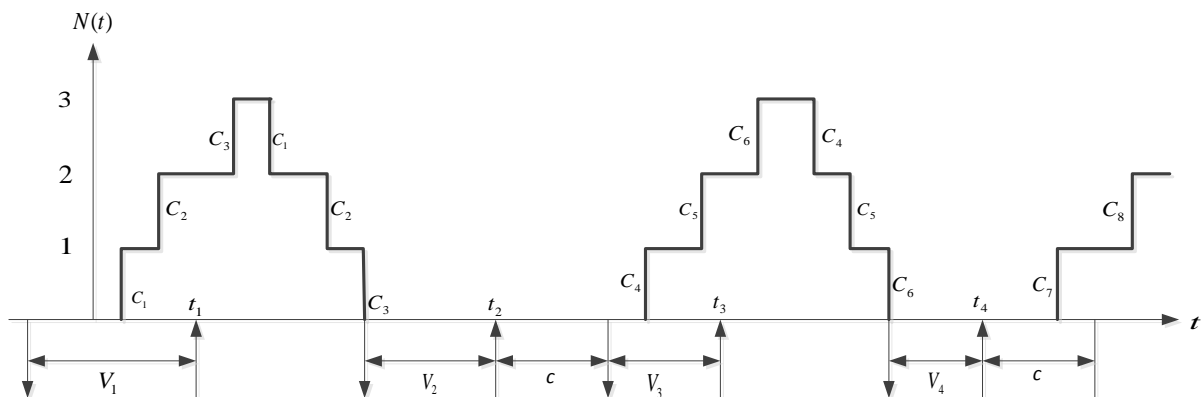


Figure 1. Vacation queueing scheme with timeout.

expiration of the timeout rather than wait indefinitely. Thus, when the source is subject to breakdown, such as the disruption of the communication links along which messages arrive in a communication system, the server timeout scheme prevents the server from waiting indefinitely for customers to arrive after returning from a vacation.

Two other hybrid vacation schemes have been proposed. In Kella [21], if the server returns from the $(i-1)$ th vacation and finds the system empty, it takes another vacation with probability p_i and waits for the first customer to arrive with probability $1-p_i$. In the later case the server takes a vacation after serving exhaustively. In [4] if the server returns from a vacation and finds the system empty, it takes at most J vacations repeatedly until it finds at least one customer waiting in the system when it returns from a vacation. If no customer arrives by the J th vacation, the server waits until a customer arrives.

3. Analysis of the Model

We assume that customers arrive at the system according to a Poisson process with rate λ . The time, X , to serve a customer has a general distribution with cumulative distribution function (CDF) $F_X(x)$, mean $E[X]$ and second moment $E[X^2]$. The s -transform of the probability density function (PDF) of X , $f_X(x)$, is $M_X(s)$, which is defined by (see, for example, [22]):

$$M_X(s) = E[e^{-sX}] = \int_0^{\infty} e^{-sx} f_X(x) dx$$

where $f_X(x) = \frac{dF_X(x)}{dx}$.

The duration, V , of a vacation is also assumed to have a general distribution with CDF $F_V(v)$ and s -transform $M_V(s)$. The mean of V is $E[V]$ and its second moment is $E[V^2]$. X and V are mutually independent. Let the random variable A denote the number of customers in the system at the beginning of a busy period. The probability mass function (PMF) of A is $p_A(a) = P[A=a]$ whose z -transform, $G_A(z)$, is given by

$$G_A(z) = E[z^A] = \sum_{a=0}^{\infty} z^a p_A(a).$$

The mean of A is $E[A]$ and its second moment is $E[A^2]$. Let the random variable B denote the number of customers left behind by an arbitrary departing customer, and let L denote the number of customers in the system at an arbitrary point in time. The PMF of B is $p_B(b)$ whose z -transform is given by $G_B(z)$. The mean of B is $E[B]$ and its second moment is $E[B^2]$. Similarly, the PMF of L is $p_L(l)$ whose z -transform is given by $G_L(z)$. The mean of L is $E[L]$ and its second moment is $E[L^2]$. Let W_q denote the waiting time in the system, and let the utilization factor be defined by $\rho = \lambda E[X]$. The main result of the analysis is the following:

Theorem 1: The mean waiting time in the system is given by

$$E[W_q] = \frac{\lambda E[V^2]}{2\{[1-e^{-\lambda c}]M_V(\lambda) + \lambda E[V]\}} + \frac{\lambda E[X^2]}{2(1-\rho)}.$$

The proof of this theorem is given in the Appendix 1.

We consider the limiting cases for c :

$$\lim_{c \rightarrow 0} E[W_q] = \frac{E[V^2]}{2E[V]} + \frac{\lambda E[X^2]}{2(1-\rho)}$$

$$\lim_{c \rightarrow \infty} E[W_q] = \frac{\lambda E[V^2]}{2\{M_V(\lambda) + \lambda E[V]\}} + \frac{\lambda E[X^2]}{2(1-\rho)}.$$

These are the results obtained in [1] for the multiple vacation queueing system and the single vacation queueing system, respectively. Note that $E[W_q]$ monotonically decreases as c increases since

$$\frac{dE[W_q]}{dc} = \frac{-\lambda^2 E[V^2] M_V(\lambda) e^{-\lambda c}}{2\{[1 - e^{-\lambda c}] M_V(\lambda) + \lambda E[V]\}^2} < 0$$

which is consistent with the fact that the single vacation scheme ($c = \infty$) has a smaller mean waiting time than the multiple vacation scheme ($c = 0$). Also, in [4] it is shown that the s -transform of the PDF of the waiting time is given by

$$M_w(s) = \frac{\lambda(1-\rho)\{1 - G_A(1-s/\lambda)\}}{E[A]\{s - \lambda + \lambda M_X(s)\}}.$$

As shown in the Appendix,

$$G_A(z) = zp_2 + \frac{M_V(\lambda - \lambda z) - M_V(\lambda)}{1 - M_V(\lambda)} p_3$$

where

$$p_2 = \frac{M_V(\lambda)\{1 - e^{-\lambda c}\}}{1 - e^{-\lambda c} M(\lambda)}, \quad p_3 = \frac{1 - M_V(\lambda)}{1 - e^{-\lambda c} M(\lambda)}.$$

Applying this result to our model yields

$$M_w(s) = \frac{(1-\rho)\{\lambda[1 - M_V(s)] + s(1 - e^{-\lambda c})M_V(\lambda)\}}{\{s(1 - e^{-\lambda c})M_V(\lambda) + \lambda E[V]\}\{s - \lambda + \lambda M_X(s)\}}$$

which, on differentiation and evaluating at $s = 0$, gives the same value of $E[W_q]$ obtained earlier.

Although the model has been analyzed with fixed timeout c , the analysis can be extended to the case where the timeout is a random variable T with mean $E[T]$ and the s -transform of the PDF of T is $M_T(s)$. In this case only a slight modification is required in the results. Specifically, we replace the factor $e^{-\lambda c}$ with $M_T(\lambda)$.

Numerical Results

We assume that X is exponentially distributed with mean $E[X] = 1 \Rightarrow \lambda = \rho$. Also, if we define the PDF of X by $f_X(x) = \mu e^{-\mu x}$, $x \geq 0$, then $E[X] = 1/\mu = 1$ and $E[X^2] = 2/\mu^2 = 2$. Similarly, we assume that V has an exponential distribution with a mean of 5. This means that $f_V(v) = \gamma e^{-\gamma v}$, $v \geq 0$; $E[V] = 1/\gamma = 5 \Rightarrow \gamma = 0.2$; $E[V^2] = 2/\gamma^2 = 50$ and

$$M_V(s) = \frac{\gamma}{s + \gamma} \Rightarrow M_V(\lambda) = \frac{\gamma}{\lambda + \gamma} = \frac{0.2}{\rho + 0.2}.$$

Thus we have that

$$E[T] = E[W_q] + E[X] = \frac{25\rho(\rho + 0.2)}{0.2[1 - e^{-\rho c}] + 5\rho^2 + \rho} + \frac{\rho}{1 - \rho} + 1.$$

Table 1 shows the values of $E[T]$ for different values of c .

From the table we observe that at low values of ρ (*i.e.*, low offered load), the mean delay decreases as c increases. However, at high values of offered load, the mean delay becomes almost identical for all values of c . The reason for this is because at high values of offered load the server is likely to be serving customers more than he takes vacations.

Table 1. Values of $E[T]$ for different values of c .

ρ	$c=0$	$c=1$	$c=2$	$c=5$	$c=10$	$c=15$	$c=\infty$
0.00	6.0000	5.1666	4.5714	3.5000	2.6667	2.2500	1.0000
0.10	6.1111	5.5481	5.1379	4.3906	3.8243	3.5671	3.2540
0.20	6.2500	5.8345	5.5424	5.0492	4.7408	4.6396	4.5833
0.30	6.4286	6.1053	5.8916	5.5705	5.4176	5.3852	5.3759
0.40	6.6667	6.4062	6.2464	6.0369	5.9636	5.9539	5.9524
0.50	7.0000	6.7848	6.6631	6.5253	6.4903	6.4874	6.4872
0.60	7.5000	7.3188	7.2249	7.1331	7.1163	7.1154	7.1154
0.70	8.3333	8.1785	8.1051	8.0433	8.0351	8.0348	8.0348
0.80	10.0000	9.8660	9.8081	9.7661	9.7620	9.7619	9.7619
0.90	15.0000	14.8829	14.8369	14.8079	14.8059	14.8058	14.8058
0.95	25.0000	24.8902	24.8490	24.8249	24.8234	24.8234	24.8234

4. The N -Policy with Timeout

The N -policy was introduced by Yadin and Naor [23] and operates as follows. The server goes on vacation at the end of a busy period. The vacation ends with the arrival of the N th customer since the end of the last busy period.

The N -policy scheme with timeout operates as follows. Consider cycles of busy periods and let T_{il} denote the arrival time of the i th customer in the l th cycle, where $i = 1, 2, \dots, N$ and $l = 1, 2, \dots$. Starting from the arrival of the first customer in a given cycle, if less than $N-1$ other customers arrive within the time interval c , then the server's vacation ends and service is performed exhaustively at the end which another vacation begins. Assume that the system is empty and, therefore, the server is on the l th vacation. The vacation ends at time T_l given by

$$T_l = \min(T_{1l} + c, T_{Nl}), \quad l = 1, 2, \dots$$

That is, the vacation ends at time T_{Nl} (when the N th customer arrives) or (if less than $N-1$ other customers arrive within the timeout period c measured from when the first customer arrived), whichever comes first. This means that each busy period begins with either N customers, if $T_{Nl} \leq T_{1l} + c$, or n customers, $1 \leq n \leq N-1$, otherwise. The server serves exhaustively and takes another vacation when the system becomes empty, whatever number of customers the busy period begins with.

The scheme is illustrated in **Figure 2**, where it is assumed that $N = 3$. In the figure, during the server's first server vacation customer C_1 arrives and thus activates the timer. By the time the timeout expired only one other customer, C_2 , arrived. Since the timeout has expired with less than 3 customers in the system, the server initiates a busy period with two customers at time t_1 . Another customer, C_3 , arrived during this busy period. After serving all 3 customers, the server takes a vacation. While the server is on vacation, customer C_4 arrives to activate the timer for the next cycle. The time expires at time t_2 and the server initiates a busy period at that time with only one customer. During this busy period, customers C_5 and C_6 arrive and are served before the server commences another vacation. While the server is on vacation, customer C_7 arrives and activates the timer. Before the timer expires customer C_8 arrives. The timer expires at time t_3 and the server initiates a busy period at that time with two customers. During that busy period customer C_9 arrives and is served before the server commences another vacation. The process continues, as shown in the figure. The intervals labeled V_i , $i = 1, 2, \dots$, indicate the vacation intervals under the normal N -policy.

The analysis of the model is similar to that for the single vacation with server timeout and the mean waiting time in the system is given by the following theorem:

Theorem 2: The mean waiting time in the system is given by

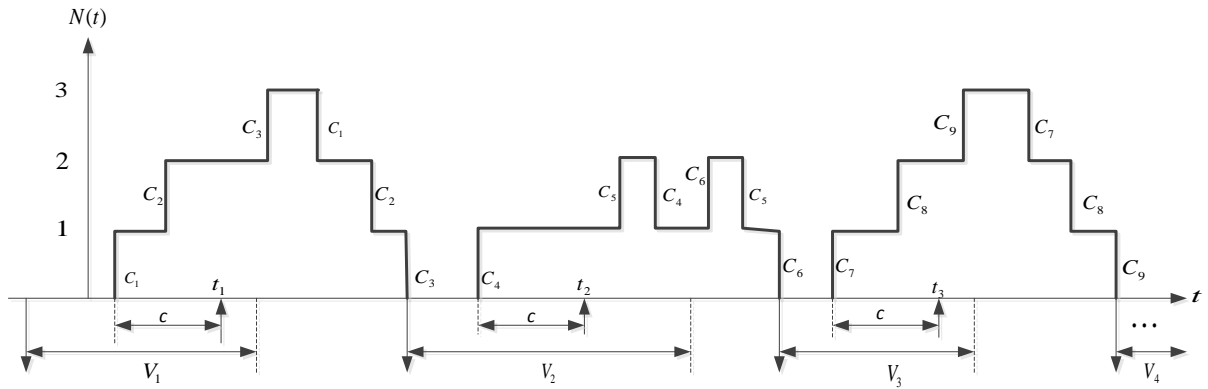


Figure 2. N -policy scheme with timeout.

$$E[W_q] = \frac{N-1}{2\lambda} + \frac{\lambda E[X^2]}{2(1-\rho)} - \sum_{n=1}^{N-1} \frac{(N-n)(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c}.$$

The proof of the theorem is given in Appendix 2 where it is also shown that

$$\lim_{c \rightarrow 0} E[W_q] = \frac{\lambda E[X^2]}{2(1-\rho)}$$

$$\lim_{c \rightarrow 0} E[W_q] = \frac{N-1}{2\lambda} + \frac{\lambda E[X^2]}{2(1-\rho)}$$

which are the results for a standard M/G/1 queue with no server vacation and a standard N -policy scheme without timeout, as shown in [23], respectively. Also, we observe that

$$\frac{dE[W_q]}{dc} = \lambda e^{-\lambda c} \left\{ \sum_{n=1}^{N-1} \frac{(N-n)(\lambda c)^{n-1}}{(n-1)!} - \sum_{n=2}^{N-1} \frac{(N-n)(\lambda c)^{n-2}}{(n-2)!} \right\} > 0$$

which means that the expected waiting time monotonically increases with c . Therefore, the optimal value of c is zero.

Numerical Results

We use the same parameters as in Section 3.1 and assume that $N = 3$. Then we have that

$$E[T] = E[W_q] + E[X] = \frac{1}{\rho} + \frac{\rho}{1-\rho} - e^{-\rho c} \{2 + \rho c\} + 1.$$

Table 2 shows the values of $E[T]$ for different values of c .

Observe that when $c \neq 0$ the minimum value of $E[T]$ occurs around $\rho = 0.5$. This can easily be verified for the case of $c = \infty$ by observing that

$$\lim_{c \rightarrow \infty} E[T] = \frac{1}{\rho} + \frac{\rho}{1-\rho} + 1.$$

Differentiating this function with respect to ρ and setting the result to zero gives the optimal value at $\rho = 0.5$. Since the second derivative is greater than zero, we conclude that $\rho = 0.5$ gives the minimum mean delay. We can also rewrite the expression for $E[T]$ as follows:

$$E[T] = \frac{1}{\rho} + \frac{\rho}{1-\rho} - e^{-\rho c} \{2 + \rho c\} + 1 = E[T]_{c \rightarrow \infty} - e^{-\rho c} \{2 + \rho c\}.$$

Table 2. Values of $E[T]$ for different values of c at $N = 3$.

ρ	$c = 0$	$c = 1$	$c = 2$	$c = 5$	$c = 10$	$c = 15$	$c = 20$	$c = \infty$
0.05	1.0526	19.1026	19.1525	19.3003	19.5363	19.7536	19.9490	21.0526
0.10	1.1111	9.2109	9.3099	9.5948	10.0075	10.3302	10.5698	11.1111
0.20	1.2500	4.4488	4.6412	5.1464	5.7086	6.0010	6.1401	6.2500
0.30	1.4286	3.0580	3.3350	3.9809	4.5130	4.6897	4.7421	4.7619
0.40	1.6667	2.5579	2.9085	3.6253	4.0568	4.1468	4.1633	4.1667
0.50	2.0000	2.4837	2.8964	3.6306	3.9528	3.9947	3.9994	4.0000
0.60	2.5000	2.7397	3.2028	3.9177	4.1468	4.1653	4.1666	4.1667
0.70	3.3333	3.4211	3.9235	4.5958	4.7537	4.7615	4.7619	4.7619
0.80	5.0000	4.9919	5.5232	6.1401	6.2466	6.2499	6.2500	6.2500
0.90	10.0000	9.9320	10.4830	11.0389	11.1097	11.1111	11.1111	11.1111
0.95	20.0000	19.9117	20.4693	20.9942	21.0518	21.0526	21.0526	21.0526

Thus, the factor $e^{-\rho c} \{2 + \rho c\}$ has a damping effect on the mean delay at $c = \infty$. This means that as c increases, the mean delay tends to slightly increase, which agrees with the statement we made earlier that the optimal value of c is zero.

Observe also that at low offered load, the value of the expected delay when $c \neq 0$ is almost equal to the value at high offered load. This is due to the delay encountered by customers by waiting for the timeout to expire before they can be served.

5. Conclusions

We have derived expressions for the mean waiting time of a vacation queueing system in which the server does not immediately take another vacation upon returning from a vacation and finding the system empty, as in the multiple vacation scheme, or wait indefinitely for a customer to arrive, as in the single vacation scheme. In the proposed model, if the server returns from a vacation and finds the system empty, it waits for a fixed time c . If at the expiration of this time, no customer arrives, the server will take a vacation; otherwise it serves arriving customers exhaustively before taking another vacation. The results of the analysis are consistent with those of the multiple vacations scheme (where $c = 0$), and the single vacation scheme (where $c = \infty$). Numerical results are also given that show the impact of the timeout period on the mean delay. A linear cost model was assumed to obtain the optimal value of c for the assumed model. Numerical results are obtained under a set of parameter assumptions, and the results indicate that there is a very small range of values of the offered load where meaningful optimization can be applied.

The model is also extended to the N -policy scheme where the timeout is measured from the arrival of the first customer since the end of the last busy period. The results have also been shown to be consistent with earlier results for the case when $c = 0$, which is the standard M/G/1 queue, and the case when $c = \infty$, which is the N -policy scheme without timeout. It is shown that the expected waiting time increases monotonically with c , which means that $c = 0$ gives the minimum expected waiting time. Numerical results are also obtained that show that the mean delay has a minimum value around $\rho = 0.5$.

References

- [1] Levy, Y. and Yechiali, U. (1975) Utilization of Idle Time in an M/G/1 Queueing System. *Management Science*, **22**, 202-211. <http://dx.doi.org/10.1287/mnsc.22.2.202>
- [2] Doshi, B.T. (1986) Queueing Systems with Vacations, a Survey. *Queueing Systems*, **1**, 29-66. <http://dx.doi.org/10.1007/BF01149327>
- [3] Doshi, B.T. (1990) Single-Server Queues with Vacations. In: Takagi, H., Ed., *Stochastic Analysis of Computer and*

Communications Systems, Elsevier, Amsterdam.

- [4] Takagi, H. (1991) Queueing Analysis: A Foundation of Performance Analysis, Volume 1: Vacation and Priority Systems, Part 1. Elsevier Science Publishers B.V., Amsterdam.
- [5] Tian, N. and Zhang, G. (2006) Vacation Queueing Models: Theory and Applications. Springer-Verlag, New York.
- [6] Fuhrmann, S.W. and Cooper, R.B. (1985) Stochastic Decomposition in M/G/1 Queue with Generalized Vacations. *Operations Research*, **33**, 1117-1129. <http://dx.doi.org/10.1287/opre.33.5.1117>
- [7] Servi, L.D. and Finn, S.G. (2002) M/M/1 Queue with Working Vacations (M/M/1/WV). *Performance Evaluation*, **50**, 41-52. [http://dx.doi.org/10.1016/S0166-5316\(02\)00057-3](http://dx.doi.org/10.1016/S0166-5316(02)00057-3)
- [8] Wu, D. and Takagi, H. (2006) M/G/1 Queue with Multiple Working Vacation. *Performance Evaluation*, **63**, 654-681. <http://dx.doi.org/10.1016/j.peva.2005.05.005>
- [9] Baba, Y. (2005) Analysis of a GI/M/1 Queue with Multiple Working Vacations. *Operations Research Letters*, **33**, 201-209. <http://dx.doi.org/10.1016/j.orl.2004.05.006>
- [10] Banik, A., Gupta, U. and Pathak, S. (2007) On the GI/M/1/N Queue with Multiple Working Vacations-Analytic Analysis and Computation. *Applied Mathematical Modelling*, **31**, 1701-1710. <http://dx.doi.org/10.1016/j.apm.2006.05.010>
- [11] Liu, W., Xu, X. and Tian, N. (2007) Stochastic Decompositions in the M/M/1 Queue with Working Vacations. *Operations Research Letters*, **35**, 595-600. <http://dx.doi.org/10.1016/j.orl.2006.12.007>
- [12] Krishnamoorthy, A. and Sreenivasan, C. (2012) An M/M/2 Queueing System with Heterogeneous Servers including One with Working Vacation. *International Journal of Stochastic Analysis*, **2012**, Article ID: 145867. <http://dx.doi.org/10.1155/2012/145867>
- [13] Ibe, O.C. and Isijola, O.A. (2014) M/M/1 Multiple Vacation Queueing Systems with Differentiated Vacations. *Modeling and Simulation in Engineering*, **2014**, Article 158247. <http://dx.doi.org/10.1155/2014/158247>
- [14] Li, J. and Tian, N. (2007) The M/M/1 Queue with Working Vacation and Vacation Interruption. *Journal of Systems Science and Systems Engineering*, **16**, 121-127. <http://dx.doi.org/10.1007/s11518-006-5030-6>
- [15] Li, J. and Tian, N. (2007) The Discrete-Time GI/Geo/1 Queue with Working Vacations and Vacation Interruption. *Applied Mathematics and Computation*, **185**, 1-10. <http://dx.doi.org/10.1016/j.amc.2006.07.008>
- [16] Li, J., Tian, N. and Ma, Z. (2008) Performance Analysis of GI/M/1 Queue with Working Vacations and Vacation Interruption. *Applied Mathematical Modelling*, **32**, 2715-2730. <http://dx.doi.org/10.1016/j.apm.2007.09.017>
- [17] Zhang, M. and Hou, Z. (2010) Performance Analysis of M/G/1 Queue with Working Vacations and Vacation Interruption. *Journal of Computational and Applied Mathematics*, **234**, 2977-2985. <http://dx.doi.org/10.1016/j.cam.2010.04.010>
- [18] Sreenivasan, C., Chakravarthy, S.R. and Krishnamoorthy, A. (2013) MAP/PH/1 Queue with Working Vacations, Vacation Interruptions and N-Policy. *Applied Mathematical Modeling*, **37**, 3879-3893. <http://dx.doi.org/10.1016/j.apm.2012.07.054>
- [19] Isijola-Adakeja, O.A. and Ibe, O.C. (2014) M/M/1 Multiple Vacation Queueing Systems with Differentiated Vacations and Vacation Interruptions. *IEEE Access*, **2**, 1384-1395. <http://dx.doi.org/10.1109/ACCESS.2014.2372671>
- [20] Ibe, O.C. and Trivedi, K.S. (1991) Stochastic Petri Net Analysis of Finite-Population Vacation Queueing Systems. *Queueing Systems*, **8**, 111-128. <http://dx.doi.org/10.1007/BF02412245>
- [21] Kella, O. (1990) Optimal Control of the Vacation Scheme in an M/G/1 Queue. *Operations Research Letters*, **38**, 724-728. <http://dx.doi.org/10.1287/opre.38.4.724>
- [22] Ibe, O.C. (2014) Fundamentals of Applied Probability and Random Processes. 2nd Edition, Elsevier Academic Press, Waltham.
- [23] Yadin, M. and Naor, P. (1963) Queueing Systems with a Removable Service Station. *Operational Research Quarterly*, **14**, 395-405.
- [24] Kleinrock, L. (1975) Queueing Systems, Volume 1: Theory. John Wiley & Sons, New York.
- [25] Little, J.D.C. (1961) A Proof of the Formula: $L = \lambda W$. *Operations Research*, **9**, 383-387. <http://dx.doi.org/10.1287/opre.9.3.383>

Appendix 1

In this section we provide a formal proof of theorem proposed in Section 3. Let $G_{L(M/G/1)}(z)$ denote the z -transform of the PMF of the number of customers in the system is a standard M/G/1 queue (*i.e.*, one in which the server never takes a vacation). As defined earlier, let $G_L(z)$ denote the z -transform of the number of customers in the system at an arbitrary point in time and let $G_B(z)$ denote the z -transform of the PMF of the number of customers left behind by an arbitrary departing customer. From the stochastic decomposition result we have that

$$G_L(z) = G_B(z)G_{L(M/G/1)}(z).$$

It is well known (see Kleinrock [24], for example) that

$$G_B(z) = \frac{1 - G_A(z)}{(1-z)E[A]},$$

$$G_{L(M/G/1)}(z) = \frac{(1-\rho)(1-z)M_X(\lambda - \lambda z)}{M_X(\lambda - \lambda z) - z}.$$

Let T denote the total time that a customer spends in the system. Thus, applying Little's law [25] we obtain the mean waiting time as

$$E[W_q] = \frac{1}{\lambda}E[T] - E[X] = \frac{1}{\lambda} \left. \frac{dG_L(z)}{dz} \right|_{z=1} - E[X] = \frac{E[A^2] - E[A]}{2\lambda E[A]} + \frac{\lambda E[X^2]}{2(1-\rho)}.$$

Thus, once $G_A(z)$ is known, we can obtain $E[W_q]$. The remainder of the proof is devoted to deriving the expressions for $G_A(z)$ and $M_W(s)$, the s -transform of the PDF of the waiting time.

Consider the following mutually exclusive events associated with the server's experience after returning from a vacation:

- 1) The server returns from vacation, waits and commences another vacation without serving a customer; the probability of this event is $M_V(\lambda)e^{-\lambda c}$.
- 2) The server returns from vacation, waits and serves at least one customer before taking another vacation; the probability of this event is $M_V(\lambda)\{1 - e^{-\lambda c}\}$.
- 3) The server returns from vacation and finds at least one waiting customer; the probability of this event is $1 - M_V(\lambda)$.

Under event 2, a busy period is initiated with exactly one customer in the system. Similarly, under event 3, a busy period is initiated with at least one customer in the system. Therefore, if we define p_k as the probability of event k , given that a busy period was initiated before the server commences another vacation, where $k = 2, 3$, then we obtain the following result:

$$G_A(z) = zp_2 + \frac{M_V(\lambda - \lambda z) - M_V(\lambda)}{1 - M_V(\lambda)} p_3$$

where

$$p_2 = \frac{M_V(\lambda)\{1 - e^{-\lambda c}\}}{1 - e^{-\lambda c}M_V(\lambda)},$$

$$p_3 = \frac{1 - M_V(\lambda)}{1 - e^{-\lambda c}M_V(\lambda)}.$$

From this we obtain the result

$$E[A] = p_2 + \frac{\lambda E[V]}{1 - M_V(\lambda)} p_2,$$

$$E[A^2] = E[A] + \frac{\lambda^2 E[V^2]}{1 - M_V(\lambda)} p_2,$$

$$E[W_q] = \frac{\lambda E[V^2]}{2E[A]\{1 - e^{-\lambda c} M_V(\lambda)\}} + \frac{\lambda E[X^2]}{2(1-\rho)} = \frac{\lambda E[V^2]}{2\{(1 - e^{-\lambda c}) M_V(\lambda) + \lambda E[V]\}} + \frac{\lambda E[X^2]}{2(1-\rho)},$$

which completes the proof.

Appendix 2

In this appendix we prove Theorem 2 that was proposed in Section 4. For the current scheme, a busy period will commence with exactly N customers in the system with probability q_N , which is the probability of $N-1$ or more Poisson arrivals in an interval of length c and thus is given by

$$q_N = 1 - \sum_{n=0}^{N-2} \frac{(\lambda c)^n}{n!} e^{-\lambda c}.$$

Similarly, a busy period will commence with n customers, $1 \leq n \leq N-1$, with probability q_n , which is the probability of exactly $(n-1)$ Poisson arrivals in an interval of length c and is given by

$$q_n = \frac{(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c}, \quad 1 \leq n \leq N-1.$$

Therefore,

$$G_A(z) = z^N q_N + \sum_{n=0}^{N-1} z^n q_n = z^N \left\{ 1 - \sum_{n=0}^{N-2} \frac{(\lambda c)^n}{n!} e^{-\lambda c} \right\} + \sum_{n=1}^{N-1} z^n \frac{(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c},$$

$$E[A] = \left. \frac{dG_A(z)}{dz} \right|_{z=1} = N \left\{ 1 - \sum_{n=0}^{N-2} \frac{(\lambda c)^n}{n!} e^{-\lambda c} \right\} + \sum_{n=1}^{N-1} \frac{n(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c} = N - \sum_{n=0}^{N-2} \frac{(N-n-1)(\lambda c)^n}{n!} e^{-\lambda c}.$$

If we define D as the event that there are N customers at the beginning of a busy period and E as the event that there are less than N customers at the beginning of a busy period, then mimicking the method used in Takagi [4] we obtain the following conditional s-transforms of the PDF of the waiting time:

$$M_W(s|D) = \frac{(1-\rho) \left\{ [\lambda/(s+\lambda)]^N - [M_X(s)]^N \right\}}{N \left\{ [\lambda/(s+\lambda)] - M_X(s) \right\}} + \frac{\lambda(1-\rho) \left\{ 1 - [M_X(s)]^N \right\}}{N \left\{ s - \lambda + \lambda M_X(s) \right\}},$$

$$M_W(s|E, n) = \frac{(1-\rho) \left\{ [\lambda/(s+\lambda)]^n - [M_X(s)]^n \right\}}{n \left\{ [\lambda/(s+\lambda)] - M_X(s) \right\}} + \frac{\lambda(1-\rho) \left\{ 1 - [M_X(s)]^n \right\}}{n \left\{ s - \lambda + \lambda M_X(s) \right\}}.$$

Thus, the unconditional s-transform of the PDF of the waiting time is given by

$$\begin{aligned} M_W(s) &= M_W(s|D) q_N + \sum_{n=1}^{N-1} M_W(s|E, n) q_n \\ &= M_W(s|D) q_N + \sum_{n=1}^{N-1} M_W(s|E, n) \frac{(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c}. \end{aligned}$$

From this we obtain the mean waiting time as

$$\begin{aligned}
E[W_q] &= -\left. \frac{dW_w(s)}{ds} \right|_{s=0} = \frac{N-1}{2\lambda} q_N + \frac{\lambda E[X^2]}{2(1-\rho)} q_N + \sum_{n=1}^{N-1} \left\{ \frac{\lambda E[X^2]}{2(1-\rho)} + \frac{n-1}{2\lambda} \right\} \frac{(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c} \\
&= \frac{N-1}{2\lambda} + \frac{\lambda E[X^2]}{2(1-\rho)} - \sum_{n=1}^{N-1} \frac{(N-n)(\lambda c)^{n-1}}{(n-1)!} e^{-\lambda c} \\
&= \frac{E[A]-1}{2\lambda} + \frac{\lambda E[X^2]}{2(1-\rho)}.
\end{aligned}$$

Before considering the limiting values of c , recall that the expected value of A is given by

$$E[A] = N - \sum_{n=0}^{N-2} \frac{(N-n-1)(\lambda c)^n}{n!} e^{-\lambda c} = N - (N-1)e^{-\lambda c} - \sum_{n=1}^{N-2} \frac{(N-n-1)(\lambda c)^n}{n!} e^{-\lambda c}.$$

Thus, $\lim_{c \rightarrow 0} E[A] = 1$ and $\lim_{c \rightarrow \infty} E[A] = N$. This implies that

$$\begin{aligned}
\lim_{c \rightarrow 0} E[W_q] &= \frac{\lambda E[X^2]}{2(1-\rho)}, \\
\lim_{c \rightarrow \infty} E[W_q] &= \frac{N-1}{2\lambda} + \frac{\lambda E[X^2]}{2(1-\rho)},
\end{aligned}$$

which are the results for a standard M/G/1 queue with no server vacation and a standard N -policy scheme without timeout, as shown in [23], respectively. This completes the proof.