

Performance Analysis of Discrete-Time Queues in Slotted Networks

G. R. DATTATREYA and LARRY N. SINGH

Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083-0688

Abstract: Many small-scale wireless networks, such as ad hoc networks, operate in a slotted mode with constant packet sizes. The transceivers have only modest hardware and hence, queuing performance is an important issue. Performance analysis of slotted systems requires more than simply obtaining the performance figures of discrete-time Markov chains. This paper elucidates the importance of characterizing the timing and synchronization details required to correctly formulate the Markov chain and evaluates corresponding performance figures. Two distinct Markov chains are identified in such analysis of any slotted system. The events causing different transitions between the system states are systematically developed for both Markov chains. Relationships between time-averaged performance figures and corresponding expectations in both Markov chains are developed. The approach is demonstrated by providing easier solutions to some slotted systems found in the literature.

Key-Words: Discrete-time queues, slotted networks, performance analysis, Markov chains, crossbar switch, timing and synchronization.

1 Introduction

Data communication progresses in a sequence of stages. Each stage consists of a subset of the following steps: arrive, wait, process, forward, and travel a physical distance. From the point of view of evaluation of the overall performance, each such stage is a queuing system. In many systems, arrivals can take place at arbitrary time instants. This is the case, for example, in a system wherein a data frame over a long distance link starts to appear at the input of a system. The processing system is digital and synchronizes the forwarding based on its own clock. In such systems the processing time is discrete, since a digital system processes in an integer number of bits. However, due to the very large number of possibilities in the number of bits processed, the processing time is justifiably approximated as a real variable. Hence, such systems are modeled and analyzed as continuous-time queuing systems. In many other small scale data communication networks, all processes proceed synchronously. Many wireless networks are organized with complete synchronization as follows. The wireless nodes synchronize their activities with the clock of one node. Channel requests, grants, data transmissions, and receptions all proceed in predetermined fixed time intervals. Data frames are necessarily of fixed sizes. Analysis and performance evaluation of such systems are crucial, especially since these systems have limited resources. A proper understanding and modeling of timing and synchronization is required to correctly analyze such discrete-time queues. It is important to be able to correctly identify when a data packet cannot enter due to a full buffer. Making assumptions with regard to synchronization issues also influences when exactly the last customer has left the system and the buffer is empty.

The vast majority of queuing theory literature per-

tains to continuous-time queuing models (Leon-Garcia [1], Trivedi [2] and Kleinrock [3], for example). The consensus is that discrete-time queues are significantly more difficult to analyze than continuous-time queues (Woodward [4]). At a given time instant, a continuous-time queue undergoes at most one state change. Compare this behaviour to a discrete-time queue which can experience multiple changes in a unit of time. This feature of discrete-time queues complicates their analysis and understanding. Nevertheless, many network systems rely on slotted-time for synchronization. Consequently, discrete-time queues have many applications in networks, such as satellite and token-passing networks. For a more thorough list of applications of discrete-time queues, see [4].

Discrete-time queues have been introduced as early as 1958 in Miesling's paper [5]. Despite the lack of emphasis given to discrete-time queues, there are a few books which treat the subject, for example Robertazzi [6], Bruneel and Kim [7], Kobayashi [8], Takagi [9] and Hunter [10]. These authors concentrate on developing mathematical techniques for the analysis and evaluation of equilibrium state probabilities for discrete parameter Markov chains, starting from various mathematical patterns for state transition probabilities.

In this paper, a simple, systematic approach to develop proper discrete-time Markov chains of slotted systems is introduced. The starting point is the physical nature of the discrete-time operation. An analysis of various statistical quantities resulting from these Markov chains is conducted. The end result is a systematic technique for the derivation of unambiguous and correct time averaged performance figures for slotted networks. Section 2 deals with the details of timing and synchronization and develops important constraints that naturally occur on the arrival and departure pos-

sibilities in empty and full buffers. Section 3 develops the two different Markov chains that result due to the choice of observation epochs in a slot. Section 4 develops the correct interpretations of various statistical quantities derived from the two Markov chains. Inter-relationships between these quantities and correct time-averaged performance figures are also derived. Section 5 presents the application of our techniques to two slotted systems found in the literature. Section 6 concludes the paper.

2 Timing and Synchronization

An important application of discrete-time queuing models is in digital data networks. In these networks, progression of activities is controlled by a clock. The clock divides time into a succession of equal intervals or *slots* (see Figure 1). These activities persist for a non-zero, finite amount of continuous time within a slot, and are simple enough that they will be completed in a small amount of time. An example of such an activity is the transmission of a packet in a wireless communication system. The end points of a slot are called *slot edges*. At the beginning of a slot, data and physical components are ready to execute an activity. By the end of a slot, the activity is complete. Even if some activity is complete before the end of the slot, the next activity cannot start until the beginning of the successive slot. Thus, the discrete-time model is not an approximation to continuous-time operation, but arises due to the strictly digital nature of the operation. In the analysis of discrete-time queuing systems, the details of the activities that take place within a slot and of the service received by the customers are unimportant. Only the numbers of customers in various positions within the system at different times are relevant. The positions of different customers should never change within the body (the open interval) of a slot in order to ensure that no activity is interrupted.

Arrival and departure events are the most common causes of changes that take place in a queuing system. Even movement of a customer within a queue is a departure from one buffer position and an arrival into another position. We make a distinction between an arrival and an arrival event (compare departure and departure event). An arrival event is the occurrence of one or more simultaneous arrivals. There is at most one arrival event per slot. Arrival and departure events (including movement of customers from one position to another) do not persist over a continuous time period, but instead are changes in the system that take place instantaneously at slot edges. Throughout this paper, unless otherwise stated, it is assumed that an arrival occurs during an infinitesimal time period *soon after* the slot edge and a departure, during an infinitesimal time period *just before* a slot edge. Figure 1 illustrates this terminology. This type of system is referred to as an *early arrival system* (EAS) in Chaudhry [11]. This model is consistent with the movement of data in digital systems, for example, in shift registers. Any activity between successive slot edges belongs to that slot. Hence, the slot edge is a natural choice of epoch to

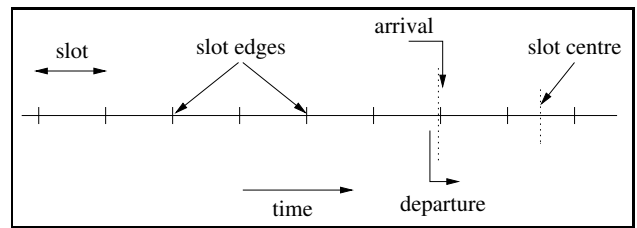


Figure 1. Slot arrival and departure instances.

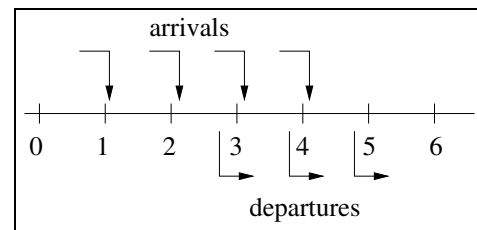


Figure 2. A Discrete Time Queuing Model showing 4 arrivals and 3 departures.

make observations. However, the numbers of customers in various positions do not change during the continuous-time period starting from soon after a slot edge and ending just before the next slot edge. Therefore, counting the numbers at *slot centres* leads to another choice of the epoch. In a practical scenario, the slot centre represents the point in the slot where arrivals are guaranteed to have fully entered the system. Compare this to the slot edge, which represents the point in a slot at which a departure is guaranteed to have completely left the system and no arrivals have begun entering the system. Due to the formulation of the system, the number of customers varies depending on whether the count is made at the slot edge or at the slot centre.

As an illustration of the difference between counting at slot edges versus counting at slot centres, let a system be empty at the beginning of slot 0, as in Figure 2 (the numbers on the time axis indicate the slot to the right of the numbers, in this figure). Let there be four successive packet arrivals in slots 1 through 4, and three successive departures in slots 2 through 4. The system has 1, 2, 2, 2, 1 packets at the centres of slots 1 through 5, respectively. At the beginning edges of slots 1 through 5 there are 0, 1, 1, 1, 1 packets in the system, respectively. At the ending edges of slots 1 through 5 there 1, 1, 1, 1, 1. The sequences of numbers of packets at the beginning edges of slots and ending edges of slots are the same except for one discrepancy: the ending edge of a slot is the beginning edge of the next slot. From the previous example, it is evident that the number in the system at any given slot depends on whether the count is made at slot edges or at slot centres. The following section derives the conditions for possible arrivals and departures in order to determine the number of customers that are in the system.

3 Formulation of the Markov chains

The state of the system at any time is the number of customers. An example of a customer is one of the packets in a store and forward queue obtaining or waiting for service. The possible set of state transitions is unambiguously determined by the epoch at which the state is evaluated. An empty state corresponds to zero customers, and the full state corresponds to the maximum number of customers possible in the system. Particular attention should be paid to the impossible transitions from empty and full states. The properties dictating these transitions are simple. Obviously, a customer cannot leave from an empty buffer and a customer cannot enter a full buffer. If the state is evaluated at the slot centre, then between the present epoch and the next, any possible departure occurs *before* any possible arrival. On the other hand, if the state is evaluated at slot edges, then any possible departure occurs *after* any possible arrival, between the present epoch and the next. Therefore, if the observations are at slot edges, then, between successive observations, the following hold. First, a customer can enter into an empty buffer and depart from the system after the slot of service. Second, a customer cannot enter a full buffer. In contrast, if observations are at slot centres, then, between successive observations, the following hold. First, a customer can enter into an empty buffer but cannot leave. Second, a customer can depart from a full buffer and another customer can enter.

The sets of events triggering possible transitions for a system which can have at most one arrival and at most one departure between the present and the next epoch are listed in Table 1. Elements in a row correspond to possible transitions. The second and third rows correspond to the cases of two different epochs. The symbol n_b denotes the size of the buffer. The event A denotes an arrival and \bar{A} denotes no arrival. Likewise, D and \bar{D} denote a departure and no departure, respectively. The symbol \cdot denotes logical AND, and $+$ denotes logical OR. The variable i in the table has the range $0 < i < n_b$. The probability of each transition is evaluated from the probability of the compound event required for the execution of the transition. These transition probabilities are functions of the arrival and service completion probabilities.

We use $\mathbf{p} = [p_{ij}]$ to denote the one step transition probabilities [12] from state i to state j , if observations are made at slot centres. If observations are made at slot edges, we use $\mathbf{q} = [q_{ij}]$ for transition probabilities from state i to state j .

Example 1: In a slotted LAN (local area network), the transmitter functions as a discrete time queue. Service consists of transmitting a packet at the head of the queue over the outgoing physical medium. Due to contentions for transmission, transmission is not always successful during an attempt. The probability of one arrival in a slot is 0.3 and the probability of two arrivals is 0.1. The capacity of the system is 3 packets. If a packet is present in the service mode of the queuing system, the probability of its service completion during the

present slot is $s = 0.6$. Probabilities of arrivals and service completions in successive slots are independent among themselves and independent of one another. An attempted arrival to a full buffer gets lost.

The arrival and departure events for different state transitions in this example are a little more involved than those in Table 1. The approach is similar and the resulting transition probability matrix for counting at slot centres is given by:

$$\mathbf{p} = \begin{bmatrix} 0.6 & 0.3 & 0.1 & 0.0 \\ 0.36 & 0.42 & 0.18 & 0.04 \\ 0.0 & 0.36 & 0.42 & 0.22 \\ 0.0 & 0.0 & 0.36 & 0.64 \end{bmatrix}. \quad (1)$$

The state transition probability matrix for counting at slot edges is:

$$\mathbf{q} = \begin{bmatrix} 0.78 & 0.18 & 0.04 & 0.0 \\ 0.36 & 0.42 & 0.18 & 0.04 \\ 0.0 & 0.36 & 0.48 & 0.16 \\ 0.0 & 0.0 & 0.6 & 0.4 \end{bmatrix}. \quad (2)$$

Evaluation of equilibrium (steady state probabilities) of all the states in a given Markov chain is a standard topic and is not repeated here. Let $\boldsymbol{\pi}_c = [\pi_c(1), \dots, \pi_c(n_b)]^T$ be the vector of equilibrium probabilities at the slot centre. Similarly, $\boldsymbol{\pi}_e = [\pi_e(1), \dots, \pi_e(n_b)]^T$ represents the vector of equilibrium probabilities at slot edges. For *Example 1*,

$$\begin{aligned} \boldsymbol{\pi}_c &= [0.264826, 0.294251, 0.253383, 0.18754]^T, \quad (3) \\ \boldsymbol{\pi}_e &= [0.441377, 0.26973, 0.213877, 0.0750159]^T. \quad (4) \end{aligned}$$

In large, complex discrete-time systems, the evaluation of equilibrium probabilities can become quite involved. Sophisticated analysis of many such cases appear in the literature (see, for example, [4] and [9]).

4 Performance Figures

4.1 Throughput

In general, different throughputs occur conditioned on different states due to state dependent arrivals or services, or both. Even in the simple *Example 1*, throughput appears to be zero when the buffer is full, if evaluated at the input. The throughput appears to be zero under the empty system condition, if evaluated at the output. Throughput is commonly evaluated by first evaluating conditional throughputs, i.e. the throughput of the system given that the system is in a particular state. The weighted average of the state dependent throughputs gives the overall throughput.

In general, a queuing system may receive simultaneous multiple arrivals soon after a slot edge as in *Example 1*. Simultaneous multiple departures may be allowed just before a slot edge if the system has multiple servers. Let a_{ij} be the probability of j arrivals being admitted when the state of the system is i at the arrival instant. Since the number of

Table 1. Events triggering various state transitions.

Transition	$0 \rightarrow 0$	$0 \rightarrow 1$	$i \rightarrow i + 1$	$i \rightarrow i$	$i \rightarrow i - 1$	$n_b \rightarrow n_b - 1$	$n_b \rightarrow n_b$
Centre	\bar{A}	A	$\bar{D} \cdot A$	$\bar{D} \cdot \bar{A} + D \cdot A$	$D \cdot \bar{A}$	$D \cdot \bar{A}$	$\bar{D} + D \cdot A$
Edge	$\bar{A} + A \cdot D$	$A \cdot \bar{D}$	$A \cdot \bar{D}$	$\bar{A} \cdot \bar{D} + A \cdot D$	$\bar{A} \cdot D$	D	\bar{D}

arrivals that can be admitted into the buffer cannot exceed the number of vacant positions in the buffer,

$$a_{ij} = 0, i + j > n_b. \quad (5)$$

Also,

$$\sum_{j=0}^{n_b-i} a_{ij} = 1, \quad i = 0, \dots, n_b - 1 \text{ and} \quad (6)$$

$$a_{n_b 0} = 1. \quad (7)$$

Similarly, let s_{ij} be the probability of j departures given that the state of the system is i at the time of departure (i.e. at the slot centre).

$$s_{ij} = 0, \quad j > i, \quad (8)$$

since there cannot be more departures than there are customers in the system. Of course,

$$\sum_{j=0}^i s_{ij} = 1. \quad (9)$$

In practice, the probabilities of arrivals that are actually admitted into the system are state dependent, and the same is true for departure probabilities. Furthermore, the state transition diagram of such a Markov will have arcs (or arrows) between non-adjacent states. Consequently, calculation of both a_{ij} and s_{ij} requires some care and thought. The throughput, nonetheless, can be easily expressed in general form as

$$Y = \sum_{i=0}^{n_b-1} \pi_c(i) \left[\sum_{j=1}^{n_b-i} j a_{ij} \right] \quad (10)$$

if evaluated by considering customers admitted into the system. If evaluated by considering all the departures, the same throughput is evaluated as

$$Y = \sum_{i=1}^{n_b} \pi_e(i) \left[\sum_{j=1}^i j s_{ij} \right]. \quad (11)$$

For *Example 1*, $Y = 0.4411$.

4.2 Buffer Occupancy

The buffer occupancy is the time average of the number of customers in the system and is a very important performance criterion. Buffer occupancy is denoted by $E[N]$ and is evaluated as the weighted average of the buffer occupancies at different states. The weights are the steady state probabilities of different states in the system. Again, we have two different Markov chains giving us two different sets of steady state probabilities requiring some resolution. The expected numbers of customers at the slot centre and at the slot edges are clearly two distinct quantities representing the physical averages of the number of customers observed at the two different epochs. Hence,

$$E[N_c] = \sum_{i=1}^{n_b} i \pi_c(i) \text{ and} \quad (12)$$

$$E[N_e] = \sum_{i=1}^{n_b} i \pi_e(i). \quad (13)$$

For *Example 1*, $E[N_c] = 1.3636$ and $E[N_e] = 0.9225$. These results lead us to the following questions. Which is the more useful quantity? Which (if either) represents the true time average of buffer occupancy? Finally, which should be used in the Little's result to evaluate the response time? In reality, the overall time average of the number of customers is

$$\frac{1}{\tau} \int_{t=0}^{\tau} E[N(t)] dt \quad (14)$$

where $E[N(t)]$ is the expected number of customers at the real variable time t and $(0, \tau]$ is the time period of one slot. However, in our ideal discrete-time queue, arrivals occur in an infinitesimal time after the slot edge and departures occur during an infinitesimal time before a slot edge. Thus, $E[N_c]$ is the expected number of customers for the entire slot time not including the slot edge points. The $E[N_e]$ is the expected number of customers for only an infinitesimal time period during a slot and hence, its contribution vanishes in the above integral. A by-product of this discussion is that the difference $E[N_c] - E[N_e]$ is the average number of customers that leave the system just before a slot edge. Of

course, the same difference also represents the average number of customers that arrive soon after a slot edge. Hence,

$$E[N_c] - E[N_e] = Y. \quad (15)$$

For *Example 1*, $E[N_c] = 1.3636$, $E[N_e] = 0.9225$ and the difference equals Y .

4.3 Response Time.

The average response time is the expected number of slots spent by a customer in the system, and is denoted by $E[R]$. This is easily evaluated by using the celebrated Little's result,

$$E[R] = \frac{E[N]}{Y}, \quad (16)$$

provided we have the correct values for $E[N]$ and Y . Little's result establishes a relationship among "time averages" of the number in the system, number of arrivals per slot, and the average response time of customers. As stated in the previous section, $E[N_c]$ is the true time average of the number of customers in the system and hence, should be used in the Little's result. Therefore, $E[R] = \frac{E[N_c]}{Y}$.

4.4 Relationship between π_c and π_e

The treatment thus far, identifies the role and the physical interpretations of the various quantities in the two different Markov chains (one depicting number of customers at slot centres, and the other at slot edges). Since both Markov chains represent the same system, it is possible to work with just one Markov chain, as long as the correct interpretations are used and accounted for. In view of this observation, explicit relationships between both Markov chains are now developed. For simplicity, it is assumed that at most one arrival with probability a , and at most one service completion with probability s are allowed. The results, however, may be easily extended to allow for bulk arrivals and departures. Note that the number of customers at a slot centre is governed by the number at the preceding slot edge and any intervening arrival. Similarly, the number of customers at a slot edge is governed by the number at the preceding slot centre and any intervening departure. Hence,

$$\pi_c(0) = \pi_e(0)(1 - a) \quad (17)$$

$$\pi_c(i) = \pi_e(i - 1)a + \pi_e(i)(1 - a), \quad (18)$$

$$\pi_c(n_b) = \pi_e(n_b) + \pi_e(n_b - 1)a, \quad (19)$$

and

$$\pi_e(0) = \pi_c(0) + \pi_c(1)s \quad (20)$$

$$\pi_e(i) = \pi_c(i)(1 - s) + \pi_c(i + 1)s, \quad (21)$$

$$\pi_e(n_b) = \pi_c(n_b)(1 - s), \quad (22)$$

where $0 < i < n_b$ in equations (18) and (21).

5 Comparative Examples

5.1 Slotted Crossbar

A very simple example that clearly illustrates the advantages of considering the two different Markov chains and selecting the one that is better suited for the problem at hand is the "Output Queueing in a Space-Division Packet Switch," or crossbar switch studied by Karol, Hluchyj, and Morgan [13]. The slotted system has N input lines and N output lines. Packets appear at the inputs of each line with an independent and identical probability p . Each packet is required to be forwarded to one of the output lines. For each packet, all the destination lines are equally likely. The output queued system functions as follows. At the beginning of a slot, the hardware for the output line under consideration (which is referred to as line A) quickly scans all the input lines, picks the packets meant for the output line A , and drops all those packets into a queuing buffer (also called buffer A here). The server of this queue forwards exactly one packet, if one or more are available in the buffer. The forwarded packet departs from the buffer at the end of the slot. Therefore, every packet spends at least one slot in the crossbar. One of the quantities under study is the number of packets left over at the end of the slot, for a large N . These are the packets that spend longer than the minimum one slot of life time in the crossbar. The number of arrivals during a slot into the buffer A is very well approximated as a Poisson random variable with a mean number of p . All such arrivals can be considered to be dropped instantaneously into the buffer right at the beginning of the slot, since the packets spend all of the slot in the system, and one packet leaves the system at the end of the slot, irrespective of the exact time instant during the slot that the packet is dropped into the buffer. Let the probability of k of these arrivals be a_k .

Karol *et al* [13] construct the discrete parameter Markov chain for the number of packets left at the end of the slot. They point out that these are the "packets in the waiting line." The transition probabilities in their Markov chain are

$$q_{ij} = a_{j-i+1}, \text{ if } i > 0 \text{ and } j \geq -1 \quad (23)$$

$$q_{00} = a_0 + a_1 \quad (24)$$

$$q_{0j} = a_{j+1}, \text{ if } j > 0. \quad (25)$$

Karol *et al* [13] obtain the solution for the equilibrium state probabilities with considerable effort.

Instead of using Markov chain at the slot edge, due to Karol *et al* [13], the Markov chain at the slot centre is formulated here; the following are the transition probabilities,

$$p_{ij} = a_{j-i+1}, \text{ if } i > 0 \text{ and } j \geq -1 \quad (26)$$

$$p_{0j} = a_j, \text{ } j \geq 0. \quad (27)$$

The transition probabilities in equations (26) and (27) correspond exactly to those in the standard M/G/1 queuing system [6]. All the necessary quantities such as the equilibrium probabilities of the number of packets in the system, their

expected number, the expected response time (including the exact one slot service time), etc. are readily obtained by using p as the arrival rate, and a constant of 1 slot for service time with zero variance, in the widely available results for the standard M/G/1 queuing system. Indeed, these figures are valid for all the times during the body of the slot. At the slot edges, the expected number momentarily dips by the throughput p before the arrivals occur. The expected number of slots in waiting is the expected response time minus the one slot of service time.

5.2 Late Arrival Systems

In our system presented here, each packet is required to be in service for at least one slot. This is typical of every synchronous, electronic hardware. Chaudhry [11] and Hunter [10] refer to the following variation called the *late arrival system* (LAS). In LAS, any arrival occurs just before the slot edge and any departure occurs soon after the slot edge. This can easily be accommodated in our Markov chain at the slot centre by simply disallowing an arrival when the buffer is full. All other aspects of the Markov chain and performance evaluations remain the same as in our development. LAS systems can allow for instantaneous service completion, and such systems are referred to as LAS-IA (Immediate Access). LAS systems which enforce that an arriving packet must wait for a minimum of one slot, even if the service facility is free, are referred to as LAS-DA (Delayed Access). The distinction between LAS-IA and LAS-DA systems manifests itself in the definition of the service time probabilities. An example of an LAS system with an unlimited buffer is in Gao *et al* [14]. Although the count in [14] is made at slot edges, counting of all arrivals occurs just before the slot edge. Thus, the count of packets is the number in the body of the slot (slot centre) and not at the time instant between departures and arrivals.

6 Conclusion

In slotted systems, the state can be observed at two distinct epochs during a slot. This possibility leads to two distinct Markov chains that represent the same physical system. These Markov chains, one observed at slot centres and the other at slot edges, are inter-related. Indeed, the steady state probabilities of one of these chains can be determined with the help of those of the other chain. This is a useful feature, since expectations from both chains are required to evaluate the common performance figures.

This paper has concentrated on the proper use of timing and synchronization to help with the formulation of Markov chains for slotted systems. The relevant performance figures, including throughput, buffer occupancy and expected response time are developed using the equilibrium probabilities from the appropriate Markov chains. The methods presented are applied to a simple, illustrative example based on slotted LANs, a second example stemming from the crossbar switch studied in [13], and a third, the LAS system. The corresponding performance figures are

also evaluated for these examples. The approach given here is quite simple to follow and easily extended to more complex scenarios. The end result is a systematic, coherent method for performance analysis of slotted systems.

References

- [1] A. Leon-Garcia, *Probability and random processes for electrical engineering*. MA, U.S.A.: Addison-Wesley, 1994.
- [2] K. S. Trivedi, *Probability and Statistics with Reliability Queuing and Computer Science Applications*. NY, U.S.A.: John Wiley & Sons, 2002.
- [3] L. Kleinrock, *Queuing Systems*. NY, U.S.A.: John Wiley & Sons, 1974.
- [4] M. E. Woodward, *Communication and computer networks: modeling with discrete-time queues*. CA, U.S.A.: IEEE Computer Society Press, 1994.
- [5] T. Miesling, "Discrete-time queuing theory," *Operations Research*, no. 6, pp. 96–105, 1958.
- [6] T. G. Robertazzi, *Computer Networks and Systems: Queuing Theory and Performance Evaluation*. NY, U.S.A.: Springer-Verlag, 1994.
- [7] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems*. MA, U.S.A.: Kluwer Academic, 1993.
- [8] H. Kobayashi, "Discrete-time queueing systems," in *Probability Theory and Computer Science* (G. Louchard and G. Latouche, eds.), ch. 4, pp. 53–121, CA, U.S.A.: Academic Press Inc., 1983.
- [9] H. Takagi, *Queueing Analysis - A Foundation of Performance Evaluation: Volume 3, Discrete-Time Systems*. NY, U.S.A.: North Holland, 1993.
- [10] J. J. Hunter, *Mathematical Techniques of Applied Probability*, vol. 2. CA, U.S.A.: Academic Press Inc., 1983.
- [11] M. Chaudhry, "On numerical computations of some discrete-time queues," in *Computational Probability* (W. K. Grassmann, ed.), ch. 10, pp. 365–408, MA, U.S.A.: Kluwer Academic, 2000.
- [12] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. NY, U.S.A.: Van Nostrand, 1960.
- [13] M. J. Karol, M. G. Hluchy, and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Transactions on Communications*, vol. COM-35, no. 12, pp. 1347–1356, 1987.
- [14] P. Gao, S. Wittevrongel, and H. Bruneel, "Discrete-time multiserver queues with geometric service times," *Computers and Operations Research*, vol. 31, pp. 81–99, 2004.