Descendant Set: An Efficient Approach for the Analysis of Polling Systems

Alan G. Konheim, Hanoch Levy, *Member, IEEE*, and Mandyam M. Srinivasan

Abstract - Polling systems have been used to model a large variety of applications and much research has been devoted to the derivation of efficient algorithms for computing the delay measures in these systems. Recent research efforts in this area, which have focused on the optimization of these systems, have raised the need for very efficient such algorithms.

This work develops the descendant set approach as a general efficient algorithm for deriving all moments of customer delay (in particular, mean delay) in these systems. The method is applied to a very large variety of model variations, including: 1) The exhaustive and gated service policies, 2) Fractional service policies, 3) The cyclic visit order, 4) Arbitrary periodic visit orders (polling tables), and 5) Customer routing. For most of these variations the method significantly outperforms the algorithms commonly used today.

I. INTRODUCTION

A large variety of computer and communications systems consisting of N stations which share a single resource are modeled and analyzed via the polling system model. The most common application is the Token Ring network in which N stations attempt to transmit their messages by sharing a single transmission line using cyclic polling mechanism. Due to the importance of these models their analysis focusing on deriving the customer (message) delay, received much attention in the last two decades. Recent research trends in this area (e.g., Boxma, Levy and Weststrate [1,2]) which have been focusing on the *optimization* of these systems have raised the need for *very efficient* algorithms for their delay analysis.

Several numerical procedures have been proposed in the past for the derivation of the mean delay (and other delay moments) in various polling systems with gated-type or exhaustive-type service. Among these procedures, the most efficient ones are applicable only to a limited number of polling system variations. Specifically, the approaches and their main characteristics are: 1) *The Buffer Occupancy Approach*; this approach is based on computing the moments of the buffer occupancy at polling instants. The method has been applied almost to all variations of polling systems and it requires $O(N^3 \log_\rho \varepsilon)$ steps (where ρ is the system utilization and ε is the relative accuracy required) to compute the N mean delay figures. 2) *The Station Time Approach*; an iterative method which has been applied to a limited number of polling systems and requires $O(N^2 \log_\rho \varepsilon)$ steps to compute the N mean delay figures. 3. *Sarkar and Zangwill's [3] Approach*; this method requires the solution of N linear (and non-sparse) equations; it requires $O(N^3)$ steps to compute the N mean delay figures and has been applied only to cyclic systems with either exhaustive or gated service. 4. *Swartz's [4] Approach*; The method uses the buffer occupancy variables and has been applied only to the discrete time exhaustive cyclic system. It requires $O(N \log_\rho \varepsilon)$ steps to compute the mean delay of a single station.

The objective of this work is to derive a general efficient computation method which can be applied to all polling system variations in which the service is of the exhaustive-type or gated-type[1]. We develop the *descendant set approach* which is based on counting the number of descendants generated in the system by each customer. We apply the method to *all* variations of polling systems which are based on fixed order of visit and use either exhaustive-type or gated-type service and derive an upper bound for the amount of computation it requires. We demonstrate that with a little effort the method can be used to derive not only the mean delay but also second and higher delay moments. The method is applied to continuous time models but can be transformed to discrete time (slotted) models. In the specific case of the discrete-time cyclic-visit exhaustive-service system, it coincides with Swarz's [4] analysis.

As shown in this work, the descendant set approach is superior to the other commonly used approaches due to its low computational complexity. For example, the derivation of the mean delay in a single station in a cyclic polling system with N stations requires $O(N \log_\rho \varepsilon)$ elementary (addition and multiplication) operations; for comparison, the Sarkar and Zangwill approach requires for this system $O(N^3)$ operations for solving the mean delay in all N stations. The

Paper approved by Hideaki Takagi, the Editor for Queueing and Networking Performance of the IEEE Communications Society. Manuscript received February 5, 1992; revised June 9, 1992. The work of A.G. Konheim and H. Levy was partially supported by the American-Israeli Binational Science Foundation grant 88-00055. Conference Information: A partial and preliminary result of this work has been reported in A.G. Konheim and H. Levy, "Efficient Analysis of Polling Systems", INFOCOM' 92.

Alan G. Konheim is with the Department of Computer Science, University of California Santa-Barbara, Santa-Barbara, CA.

Hanoch Levy is with RUTCOR, Rutgers University, P.O.Box 5062, New Brunswick, NJ 08903-5062 (on leave from Tel-Aviv University).

Mandyam M. Srinivasan is with the Management Science Program, College of Business Administration The University of Tennessee, Knoxville, TN 37996.

IEEE Log Number 9401023.

[1] These do not include systems with limited-type service, like the limited-1 or the Bernoulli service, whose analysis is usually very computationally demanding; see, e.g., Servi [5], Leung[6], Leung and Eisenberg [7], Tedijanto [8], and LaMaire [9] for these models.

main advantages of the descendant set approach are:

1) The delay computation of one station is independent of that of the others. For this reason, the superiority of the method is mostly expressed when not all N delay figures are needed. This is often the case in large systems in which various stations can be considered to be similar to each other, and thus not every one of them requires separate delay analysis.

2) The method is based on the buffer occupancy variables and thus can be applied to most systems which have been proposed and analyzed in the past.

3) The simple structure of the expressions derived by the method opens possibilities for further analysis with potential applications in the optimization of these systems.

We present the descendant set method by applying it to the derivation of the mean delay in the cyclic-visit system with either gated or exhaustive service (Section III). We then show (Section IV) that it is rather simple to apply it to a variety of models (described in Section II) and performance measures: 1) Mean delay in a system with periodic polling (polling table), 2) Mean delay in a system with fractional service (e.g., binomial-gated service), 3) Mean delay in a system with customer routing and customer branching, and 4) Second moment (and higher moments of) the delay in all these systems. For all these systems we derive upper bounds for the number of operations required by the method and compare the efficiency of the method to that of the current common techniques (Section V).

II. Model Description

We consider in this work several models which we describe below.

A. The Basic Model: Cyclic Polling with (Pure) Gated or (Pure) Exhaustive Service

The system consists of a single server and N infinite buffer queues (stations), denoted $Q_1, \cdots, Q_N$. The customer arrivals to Q_i form an independent Poisson stream. Customers arriving at Q_i are called type-i customers. The service time of a type-i customer is an independent random variable B_i with Laplace-Stieltjes Transform (LST) $B_i^*(s)$ (namely, $B_i^*(s) = Ee^{-sB_i}$), and first three moments b_i, $b_i^{(2)}$ and $b_i^{(3)}$. The utilization of Q_i is $\rho_i = \lambda_i b_i$; the overall system utilization is $\rho = \sum_{i=1}^{N} \rho_i$. Let Θ_i be the duration of a busy period in an M/G/1 system with the arrival rate λ_i and service time B_i; let $\Theta_i^*(s)$ be the LST of Θ_i.

The service at each queue is either according to the *gated* policy or the *exhaustive* policy. In the gated policy when the server visits Q_i it serves all the customers it finds there upon its arrival (but does not serve customers which arrive during the service period of Q_i). In the exhaustive policy when the server visits Q_i it serves the customers there until the queue is empty. It is well known (see e.g., Takagi [10]) that under the gated or the exhaustive service the system is stable as long as $\rho < 1$.

After completing the service of Q_i, the server incurs a switch-over period whose duration is an independent random variable, R_i, whose LST and first three moments are $R_i^*(s)$, r_i, $r_i^{(2)}$ and $r_i^{(3)}$. The period at which the server serves Q_i is called a *service period* of Q_i and the switch-over period succeeding it is called a *switch-over period* of Q_i. The instant at which the service period of Q_i starts is called a *polling instant* of Q_i. The visit order (which is the order in which the server visits the queues) in the basic model is cyclic and thus the system is called a *cyclic polling system.* The reference to queue index in this model is done modulo N.

The delay incurred by an arbitrary customer at Q_i is denoted W_i. The main performance measure of interest is the mean value of the delay experienced by type-i customers, namely, $E[W_i]$, $i=1, \cdots, N$. Other important performance measures are the second moment (or, equivalently, the variance) of the delay at Q_i, $E[W_i^2]$, and higher moments of the delay.

B. Model Variations

1) Polling According to a Polling Table: In this system the visit order is not necessarily cyclic but rather periodic with a general pattern. To define the visit order one uses a table T containing M $(M \geq N)$ entries representing the M visits to be conducted in a cycle. The value of $T[i]$ is the index of the queue visited in the ith visit in the cycle (pattern), thus $T[i] \in \{1, \cdots, N\}$. Polling tables can be used to effectively prioritize the different queues and thus are important for optimization purposes. The analysis of the mean delay in polling systems with polling tables is provided in Eisenberg [11] (using the buffer occupancy approach) and Baker and Rubin [12] and Choudhury [13] (using the station time approach). The former method requires $O(MN^2 \log_\rho \varepsilon)$ elementary operations for computing the N mean delay figures while the latter requires $O(M^2 \log_\rho \varepsilon)$ operations.

2) Polling According to Markovian (Random) Polling: In this system the visit order is not periodic but rather Markovian (or random). In the *Markovian* system, after visiting Q_i, $i = 1, \cdots, N$, the server incurs a type-i switch-over period and then goes to serve Q_j, $j = 1, \cdots, N$, with probability $p_{i,j}$. The *random* polling system is a specific case of the Markovian system in which $p_{i,j} = p_j$, $i = 1, \cdots, N$, $j = 1, \cdots, N$. Random and Markovian polling systems have been introduced and analyzed by Kleinrock and Levy [14] and Boxma and Weststrate [15], respectively. Srinivasan [16] studies the nondeterministic polling system, which generalizes the Markovian polling system to the case where the routing probability and switchover times may depend on whether a service was performed prior to the switch.

3) Fractional Service Policies: Fractional service policies are exhaustive-type and gated-type policies equipped with N parameters, q_i, $i = 1, \cdots, N$, obeying $0 < q_i \leq 1$ and used to prioritize the queues by controlling the amount of service given to each of them during a visit of the server. Several such policies have been introduced in the recent years: 1) The *binomial gated (fractional gated)* policy (Levy [17]), in which the number of customers served in a visit to Q_i is determined by the binomial distribution with parameters X_i^i (the number of customers present at Q_i when it is polled) and q_i; 2) The *binomial exhaustive* (Boxma [18] and Browne and Yechiali [19]) and 3) The *fractional exhaustive* policy (Levy [20]) which are exhaustive-type extensions of the binomial-gated policy.

4) Customer Routing: The customer routing model is a generalization of the basic polling model. In this system, upon service completion at Q_i, the customer served does not necessarily leave the system (as it would do in the basic system) but rather may be routed to one of the system queues. The routing rules are probabilistic and a type-i customer will be routed to Q_j with probability $p_{i,j}$, $j = 1,...,N$, and will leave the system with probability $p_{i,0}$; these probabilities obey $\sum_{j=0}^{N} p_{i,j} = 1$. Once moved to Q_j, the customer now becomes a type-j customer. The delay analysis of this system is provided in Sidi, Levy and Fuhrmann [21]. A further generalization of this model allows a customer to "branch off" and to send several offsprings to several queues upon service completion at Q_i. The feature of customer routing significantly extends the modeling capabilities of polling systems.

III. The Analysis

A. A Review of the Buffer Occupancy Variables and the Buffer Occupancy Approach

Let X_i^j be the number of customers present at Q_j at a polling instant of Q_i, when the system is in equilibrium. The buffer occupancy variables are the set $\{X_i^j\}_{1\le i,j\le N}$ and the buffer occupancy analysis approach is based on computing the moments of these variables. The most important moments are $E[X_i^j]$ and $E[X_i^jX_i^k]$, $1 \le i, j, k \le N$, which are required for deriving the mean delay in the system. Similarly, higher occupancy moments, e.g., $E[X_i^jX_i^kX_i^l]$ can be defined.

The main principle of the buffer occupancy approach is to follow the evolution of the system in forward direction and to compute the set $\{E[X_i^jX_i^k]\}_{1\le j,k\le N}$ from the set $\{X_{i-1}^jX_{i-1}^k\}_{1\le j,k\le N}$ using a simple (though large) set of linear equations. The derivation of the mean delay at Q_i can be done easily from the values $E[X_i^i]$ (for which a closed form expression is readily available) and $E[(X_i^i)^2]$. For example, the mean delay at the cyclic polling system with gated service is given by

$$EW_i \;=\; (E[(X_i^i)^2]-E[X_i^i])(1+\rho_i)/(2\lambda_i E[X_i^i]).$$

Higher moments of the delay can be derived using the higher moments of the buffer occupancy in a similar manner.

B. The Principles of the Descendant Set Approach

We classify the customers of a polling system into two classes: 1) *Original customers (originators)*, and 2) *Non-original customers.* An original customer is a customer which arrives at the system during a switch-over period. A non-original customer is a customer which arrives at the system during the service of another customer (which itself can be either an original or a non-original customer). Let C be a customer, then the *children set* (or, *immediate descendant set*) of C is the set consisting of the customers arriving to the system during the service of C; the *descendant set* of C is recursively defined to consist of C, its children (if any) and the descendants of its children. We say that C' *belongs* to C if C' is in the descendant set of C and C is an originator. The use of descendant sets in the analysis of polling systems and M/G/1 systems can be found also in Fuhrmann and Cooper [22], Boxma and Groenendijk [23] and Coffman and Stoylar [24].

In the analysis described below we focus on the derivation of the mean delay at Q_1. Obviously, one can apply this analysis to any other queue by a simple renaming of the queue indices. Similarly to the buffer occupancy approach, the objective of the descendant set analysis is to derive the moments of the buffer occupancy at Q_1 at a polling instant of that queue. The main idea behind this approach is the observation that each of the X_1^1 customers present at Q_1 when it is polled belongs to exactly one originator which arrived in one of the past switch-over periods. We therefore concentrate on an arbitrary originator C_i that arrived at Q_i in the past and calculate the number of its type-1 descendants present at Q_1 when it is polled. Summing up these numbers over all past originators C will obviously yield X_1^1.

We look at the Markov chain embedded at the polling instants of the system. Our *reference point* is a polling instant of queue 1, and we consider all the polling instants prior to this instant. Let $S_{i,0}$ be the service period of Q_i at the cycle preceding the reference point, $S_{i,1}$ be the service period of Q_i at the cycle preceding that cycle and so on. In general, we denote by $S_{i,c}$, $i=1, \cdots, N$, $c=0,1,2, \cdots$, the service period of Q_i that occurred c cycles ago. Let $C_{i,c}$ denote a type-i customer considered for service during the service period $S_{i,c}$. Let $L_{i,c}$ be a random variable denoting the number of type-1 customers which are descendants of $C_{i,c}$ and which are present at queue 1 at the *reference point.* $L_{i,c}$ can be viewed as the "contribution" of $C_{i,c}$ to X_1^1. We let $L_{i,c}(z)$ be the Generating Function (GF) of $L_{i,c}$ (namely, $L_{i,c}(z) = Ez^{L_{i,c}}$) and $\alpha_{i,c} = E[L_{i,c}]$, $\alpha_{i,c}^{(2)} = E[(L_{i,c})^2]$ and $\alpha_{i,c}^{(3)} = E[(L_{i,c})^3]$.

C. Analysis of the Gated Service Policy

To compute the GF and the moments of $L_{i,c}$ we note that the type-1 descendants of $C_{i,c}$ present at Q_1 at the reference point are the type-1 descendants of the children of $C_{i,c}$. The number of children of $C_{i,c}$ is obviously the number of Poisson arrivals during its service (whose duration is B_i). Noting that in the gated system, for every $j>i$ all type-j customers which arrive during the service of $C_{i,c}$ are served in the current cycle (and thus the contribution of each of them is $L_{j,c}(z)$) while for every $j\le i$ all type-j customers which arrive during the service of $C_{i,c}$ are served in the next cycle (and thus the contribution of each of them is $L_{j,c-1}(z)$), we get:

$$L_{i,c}(z) = B_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,c}(z)] + \sum_{j=1}^{i}[\lambda_j-\lambda_j L_{j,c-1}(z)]\right] \quad c\geq 1 \qquad (3.1a)$$

For $c=0$ this expression slightly changes since customers that are to be served after the reference point do not contribute to X_1^1. Thus, we have:

$$L_{i,0}(z) = B_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,0}(z)] + [\lambda_1-\lambda_1 z]\right] \qquad (3.1b)$$

For compactness of presentation we extend the definition of $L_{i,c}$, $\alpha_{i,c}$, $\alpha_{i,c}^{(2)}$, $\alpha_{i,c}^{(3)}$ to allow $c=-1$ which is the cycle just past the reference point; obviously we have $L_{1,-1}=1$, $L_{j,-1}=0$ $(2\leq j\leq N)$ (deterministically) and thus $\alpha_{1,-1}=\alpha_{1,-1}^{(2)}=\alpha_{1,-1}^{(3)}=1$, and $\alpha_{j,-1}=\alpha_{j,-1}^{(2)}=\alpha_{j,-1}^{(3)}=0$, $2\leq j\leq N$. From (3.1a) and (3.1b) we may now derive a simple recursive equation for the moments of $L_{i,c}$:

$$\alpha_{i,c} = b_i\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c} + \sum_{j=1}^{i}\lambda_j\alpha_{j,c-1}\right] \quad c\geq 0 \qquad (3.2)$$

$$\alpha_{i,c}^{(2)} = b_i^{(2)}\cdot\left[\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c} + \sum_{j=1}^{i}\lambda_j\alpha_{j,c-1}\right]^2\right] + b_i\cdot\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c}^{(2)} + \sum_{j=1}^{i}\lambda_j\alpha_{j,c-1}^{(2)}\right] \quad c\geq 0 \qquad (3.3)$$

Let us now compute the number of type-1 descendants belonging to *all the originators* which arrive at the switch-over period succeeding $S_{i,c}$. Let us denote this number by $R_{i,c}$ and its GF by $R_{i,c}(z)$, and let $\beta_{i,c} = E[R_{i,c}]$, $\beta_{i,c}^{(2)} = E[(R_{i,c})^2]$, $\beta_{i,c}^{(3)} = E[(R_{i,c})^3]$. Note that $R_{i,c}$ can be viewed as the "contribution" of the switch-over period succeeding $S_{i,c}$ to X_1^1. Then, we have:

$$R_{i,c}(z) = R_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,c}(z)] + \sum_{j=1}^{i}[\lambda_j-\lambda_j L_{j,c-1}(z)]\right] \quad c\geq 1 \qquad (3.4a)$$

$$R_{i,0}(z) = R_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,0}(z)] + [\lambda_1-\lambda_1 z]\right] \qquad (3.4b)$$

From (3.4a-b) we may get expressions for $\beta_{i,c}^{(2)}$ and $\beta_{i,c}$. For our analysis we need the difference $\beta_{i,c}^{(2)}-\beta_{i,c}^2$, which is given by:

$$\beta_{i,c}^{(2)} - \beta_{i,c}^2 = (r_i^{(2)}-r_i^2)\cdot\left[\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c} + \sum_{j=1}^{i}\lambda_j\alpha_{j,c-1}\right]^2\right] + r_i\cdot\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c}^{(2)} + \sum_{j=1}^{i}\lambda_j\alpha_{j,c-1}^{(2)}\right] \quad c\geq 0 \qquad (3.5)$$

D. Analysis of the Exhaustive Service Policy

To analyze the exhaustive system we note that each customer which is present at the beginning of the service period of Q_i initiates a complete busy period at that queue; we denote the duration of such busy period by Θ_i. During such a busy period customers of type-j $(j\neq i)$ arrive at the system and accumulate in the other queues. The contribution of the customer initiating the busy period is accounted for by summing over the contribution of all those (type-j) customers. Note that we do not account for type-i customers which arrive during the busy period since they are all served during the busy period and do not remain in the system after it; their "contribution" is implicitly accounted for by the duration of the busy period. Thus, we may account for the descendants of $C_{i,c}$ by summing over all type-j customers $(j\neq i)$ arriving to the system during the busy period initiated by this type-i customer (whose duration is Θ_i). The equations corresponding to (3.1-3) are thus:

$$L_{i,c}(z) = \Theta_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,c}(z)] + \sum_{j=1}^{i-1}[\lambda_j-\lambda_j L_{j,c-1}(z)]\right] \quad c\geq 1 \qquad (3.6a)$$

$$L_{i,0}(z) = \Theta_i^*\left[\sum_{j=i+1}^{N}[\lambda_j-\lambda_j L_{j,0}(z)] + [\lambda_1-\lambda_1 z]\right] \qquad (3.6b)$$

$$\alpha_{i,c} = E[\Theta_i]\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c} + \sum_{j=1}^{i-1}\lambda_j\alpha_{j,c-1}\right] \quad c\geq 0 \qquad (3.7)$$

$$\alpha_{i,c}^{(2)} = E[\Theta_i^2]\cdot\left[\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c} + \sum_{j=1}^{i-1}\lambda_j\alpha_{j,c-1}\right]^2\right] + E[\Theta_i]\cdot\left[\sum_{j=i+1}^{N}\lambda_j\alpha_{j,c}^{(2)} + \sum_{j=1}^{i-1}\lambda_j\alpha_{j,c-1}^{(2)}\right] \quad c\geq 0 \qquad (3.8)$$

where $E[\Theta_i]$ and $E[\Theta_i^2]$ are the first two moments of the busy period duration in an M/G/1 queue: $E[\Theta_i] = b_i/(1-\rho_i)$, $E[\Theta_i^2] = b_i^{(2)}/(1-\rho_i)^3$. Substituting these expressions into (3.7, 3.8) we get a simple recursion for computing the sequences $\{\alpha_{i,c}\}$ and $\{\alpha_{i,c}^{(2)}\}$, $i=1,\ldots,N$, $c=0,1,\ldots$.

Obviously the relations (3.4a-b) and (3.5) still hold for this system.

E. Computational Algorithms

As stated above the objective of the algorithm is to compute the values $E[X_1^1]$ and $E[(X_1^1)^2]$. While closed form expression for $E[X_1^1]$ is well known (e.g., for gated system: $E[X_1^1] = \lambda_1 r/(1-\rho)$ where $r=\sum_{i=1}^{N}r_i$), the value of $E[(X_1^1)^2]$ needs to be numerically computed. Assuming that at the reference point the system is in equilibrium, and noting that each of customers present at Q_1 at the reference point belongs to an originator which arrived during one of the past switch-over periods, we get: $X_1^1 = \sum_{c=0}^{\infty}\sum_{i=1}^{N}R_{i,c}$, where each of the random variables in this sum is independent of the others. Thus we get:

$$E[X_1^1] = \sum_{c=0}^{\infty}\sum_{i=1}^{N}\beta_{i,c} \qquad (3.9a)$$

$$E[(X_1^1)^2] - (E[X_1^1])^2 = Var(X_1^1) = \sum_{c=0}^{\infty}\sum_{i=1}^{N}[\beta_{i,c}^{(2)}-\beta_{i,c}^2] \qquad (3.9b)$$

To efficiently compute Equation (3.9b) one needs to keep several data structures: alpha[1..N] and alpha_2[1..N] -- two N dimensional arrays for storing the N more recent values of $\lambda_i \alpha_{i,c}$ and $\lambda_i \alpha_{i,c}^{(2)}$ respectively, sum_alpha and sum_alpha_2 -- two variables for storing the sum of the N recent values of $\lambda_i \alpha_{i,c}$ and $\lambda_i \alpha_{i,c}^{(2)}$ respectively, and sum_beta_2 -- a variable representing the infinite sum expressed in Equation (3.9b).

These simple data structures allow to compute the values of $\alpha_{i,c}$, $\alpha_{i,c}^{(2)}$ and $\beta_{i,c}^{(2)}-\beta_{i,c}^{2}$ in $O(1)$ steps. To demonstrate the algorithm, let us consider its application for the gated type system:

ALGORITHM DS (for cyclic gated system)

```
for c:=0 to ∞ do begin
  for i:=N to 1 by -1 do begin
    temp := λi*bi*sum_alpha
    temp_2 := λi*(bi(2)*(sum_alpha)**2 + bi*sum_alpha_2)
    sum_beta_2 := sum_beta_2 + (ri(2)-ri2)*(sum_alpha)**2
                  + ri*sum_alpha_2
    sum_alpha := sum_alpha + temp - alpha[i]
    sum_alpha_2 := sum_alpha_2 + temp_2 - alpha_2[i]
    alpha[i] := temp;
    alpha_2[i] := temp_2;
  end;
end;
```

The variables are to be initialized at: alpha[i] = alpha_2[i] = 0 $(2 \le i \le N)$ and alpha[1] = alpha_2[1] = sum_alpha = sum_alpha_2 = λ_1. The external loop of this algorithm should run until the convergence of sum_beta_2 is achieved.

To evaluate the convergence time of this algorithm note that the value computed by the algorithm after performing k external iterations is equal to the second moment of X_1^1 in a system that starts operating empty of customers and runs for k cycles. Thus, the convergence of this algorithm is identical to that of the buffer occupancy approach (see Levy [25]) and the number of iterations it requires to achieve relative accuracy ε is $O(\log_\rho \varepsilon)$. Thus the algorithm requires $O(N \log_\rho \varepsilon)$ elementary operations (e.g., addition, multiplication) to converge. The space required by the algorithm is $O(N)$.

For cyclic systems the algorithm can be further simplified; next we demonstrate this simplification for the gated system. Expressing $\alpha_{i,-1}^{(2)}=0$, $i \ne 1$ and $\alpha_{1,-1}=1$, we combine equations (3.2) and (3.3):

$$\alpha_{i,c}^{(2)} = b_i^{(2)} \left[\frac{\alpha_{i,c}}{b_i} \right]^2 + b_i \left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c}^{(2)} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}^{(2)} \right] \qquad c \ge 0 \qquad (3.10)$$

Now, define $\psi_i = \sum_{c=0}^{\infty} (\lambda_i \alpha_{i,c})^2$. Multiplying both sides of (3.10) by λ_i and summing over all $c \ge 0$ and over all i, and manipulation of the result yields:

$$\sum_{i=1}^{N} \sum_{c=0}^{\infty} \lambda_i \alpha_{i,c}^{(2)} = \frac{1}{1-\rho} \cdot \left[\sum_{i=1}^{N} \lambda_i b_i^{(2)} \frac{\psi_i}{\rho_i^2} + \lambda_1 \rho \right]. \qquad (3.11)$$

In a similar manner, from Equation (3.5) we get:

$$\sum_{c=0}^{\infty} \sum_{i=1}^{N} (\beta_{i,c}^{(2)} - \beta_{i,c}^{2}) = \sum_{i=1}^{N} Var(R_i) \frac{\psi_i}{\rho_i^2} + r \left[\sum_{c=0}^{\infty} \sum_{i=1}^{N} \lambda_i \alpha_{i,c}^{(2)} + \lambda_1 \right]. \qquad (3.12)$$

From (3.11) and (3.12) we get

$$\sum_{c=0}^{\infty} \sum_{i=1}^{N} (\beta_{i,c}^{(2)} - \beta_{i,c}^{2}) = \sum_{i=1}^{N} Var(R_i) \frac{\psi_i}{\rho_i^2} + \frac{r}{1-\rho} \cdot \left[\sum_{i=1}^{N} \lambda_i b_i^{(2)} \frac{\psi_i}{\rho_i^2} + \lambda_1 \right]. \qquad (3.13)$$

Thus we merely need to focus on computing ψ_i efficiently. Let $\delta_i = \rho_i/(1+\rho_i)$ and define $\Psi_{i,c} = \lambda_i \alpha_{i,c}$. Note that $\psi_i = \sum_{c=0}^{\infty} \Psi_{i,c}^2$. Expressing $\alpha_{i,-1}=0$, $i \ne 1$ and $\alpha_{1,-1}=1$, we have from (3.2): $\Psi_{i,c} = \rho_i [\sum_{j=i+1}^{N} \Psi_{j,c} + \sum_{j=1}^{i} \Psi_{j,c-1}]$, from which we can compute recursively $\Psi_{i,c}$ as follows:

$$\Psi_{N,c} = \rho_N [\Psi_{1,c-1}/\delta_1 - \Psi_{1,c-2}],$$

$$\Psi_{i,c} = \rho_i [\Psi_{i+1,c}/\delta_{i+1} - \Psi_{i+1,c-1}], \quad i < N, \qquad (3.14)$$

with $\Psi_{N,0} = \rho_N \Psi_{1,-1}$, and $\Psi_{i,0} = \rho_i \Psi_{i+1,0}/\delta_{i+1}$, $i<N$. This leads to a very simple algorithm for computing $\Psi_{i,c}$ and then ψ_i and $E[W_1]$.

In a similar manner, an efficient algorithm for deriving the mean delay in the exhaustive cyclic system can be derived.

Remark 3.1: Note that the above simplification reduces the complexity of the DS algorithm; however the reduction is only by a factor of constant.

IV. Application to Generalized Models

A. Systems with Mixed Service

The method described in Section III directly applies to systems in which some of the stations are served by the exhaustive policy and some by the gated policy. This application is done by properly embedding in the iterative procedure the exhaustive-service relations (3.7, 3.8) for exhaustive-type stations and the gated-service relations (3.2, 3.3) for gated-type stations.

B. Polling According to a Polling Table

To conduct this analysis, a cycle is defined as the sequence of M visits given according to the polling table. The queue visited at the ith visit of a cycle is called *pseudo-queue* i and is denoted PQ_i. A type-i customer now refers to a customer which is served at PQ_i. For each pseudo-queue i we define the set of N succeeding visits of PQ_i; this is a vector Z_i whose jth entry, $Z_i(j)$, denotes the first index in the table succeeding i (modulo M) in which the queue visited is Q_j. For example, if the table is T=(1,2,1,3) then $Z_1(1)$=3 $Z_1(2)$=2 and $Z_1(3)$=4. Similarly Z_2=(3,2,4) and $Z_3 = Z_4$ = (1,2,4).

We next present how the method can be used to derive the moments of buffer occupancy at $Q_{T(1)}$ when PQ_1 is polled. The application of the method is very similar to its application in the cyclic system and the change required is only in reference to indices. We demonstrate the modification by applying it to Equation (3.2):

$$\alpha_{i,c} = b_{T(i)} \left[\sum_{\substack{j=1 \\ Z_i(j)>i}}^{N} \lambda_j \alpha_{Z_i(j),c} + \sum_{\substack{j=1 \\ Z_i(j)\le i}}^{N} \lambda_j \alpha_{Z_i(j),c-1} \right] \qquad c \ge 0 \qquad (4.1)$$

Similarly, the recursions equivalent to (3.1a-b), (3.3), (3.4) and (3.5) can be obtained. The implementation of the computation algorithm described in Section III.E can be also done efficiently in a way that the computation of $\alpha_{i,c}$ requires $O(1)$ operations. As in the previous cases the convergence of this procedure is identical to that of the buffer occupancy approach and an upper bound for the number of operations required for computing the moments of a single pseudo-queue is given by $O(M \log_\rho \varepsilon)$.

C. Descendant Set and Markovian (Random) Polling

The application of the descendant set approach to random polling systems seems to be not straight forward. The difficulty in applying the method is not in recursively deriving the contributions of past switch-over periods to the population of type-1 customers at the reference point (namely, deriving relations similar to 3.1a-b, 3.2, 3.3 and 3.4a-b and 3.5). The main difficulty is in the fact that, unlike in systems with fixed visit order, the contribution of past switch-over periods are not independent of each other (since they all depend on the actual visits to be taken between their occurrence and the reference point). For this reason it remains as an open question whether the descendant set approach can be efficiently applied to Markovian (random) polling systems.

D. Fractional Service Policies

To treat fractional service policies we demonstrate the application of the method to the Binomial-gated service policy. As described in Section II, the number of customers served in a service period of Q_i is a random variable distributed according to the binomial distribution with parameters X_i^i and q_i. It is convenient in this context to view the policy as one which considers *all* X_i^i customers for service and which applies for each of them a Bernoulli experiment (with parameter q_i) regarding whether to serve it or not. This view is legitimate as long as we are concerned with counting the number of customers present at polling instants. Having this view in mind, we observe that the descendants of $C_{i,c}$ are, with probability q_i, the descendants of the arrivals during a period whose duration is B_i, and with probability $1-q_i$, the descendants of $C_{i,c-1}$. Thus Equations (3.1a-b) change to:

$$L_{i,c}(z) = q_i \cdot B_i^* \left[\sum_{j=i+1}^{N} [\lambda_j - \lambda_j L_{j,c}(z)] + \sum_{j=1}^{i} [\lambda_j - \lambda_j L_{j,c-1}(z)] \right] + (1-q_i) L_{i,c-1}(z) \qquad c \ge 0 \qquad (4.2)$$

where we define $L_{1,-1}(z)=z$ and $L_{i,-1}(z)=1$, $(2 \le i \le N)$.

The derivation of the recursive equations for $\alpha_{i,c}$ and $\alpha_{i,c}^{(2)}$ is immediate from (4.2); also, Equation (3.5) holds for this system as is. The number of iterations required for convergence is identical to that reported in Levy [17] and thus the computational complexity of this procedure is $O(N \log_a \varepsilon)$ where $a = q_{\min}\rho + 1 - q_{\min}$, and $q_{\min} = \min_i q_i$.

E. Customer Routing

The applicability of the descendant set approach to models with customer routing is demonstrated by applying it to a system with gated service, cyclic visit and customer routing.

The analysis approach is similar to that described in Section III.D, but the equations change. The equations are now based on the fact that descendants of a type-i customer which transits to Q_j (and becomes then a type-j customer) consist of the descendants of its children and the descendants of the "new" type-j customer. Thus, Equations (3.1a-b) change to:

$$L_{i,c}(z) = B_i^* \left[\sum_{j=i+1}^{N} [\lambda_j - \lambda_j L_{j,c}(z)] + \sum_{j=1}^{i} [\lambda_j - \lambda_j L_{j,c-1}(z)] \right] \cdot \left[\sum_{j=i+1}^{N} p_{i,j} L_{j,c}(z) + \sum_{j=1}^{i} p_{i,j} L_{j,c-1}(z) + p_{i,0} \right] \qquad c \ge 0 \qquad (4.3)$$

where $L_{i,-1}(z)$ is defined as in (4.2). The equations equivalent to (3.2) and (3.3) can be easily obtained by differentiation of (4.3); in addition, note that the equations for computing the switch-over period contribution (3.4a-b, 3.5) hold for this system.

Using the equations one can derive the values of $E[(X_i^i)^2]$ in $O(N \log_a \varepsilon)$ operations where a is the convergence rate provided in Sidi, Levy and Fuhrmann [21]. Note, however, that the mean delay at Q_i cannot be determined only from the values $E[X_i^i]$ and $E[(X_i^i)^2]$. Rather, it requires the derivation of all the terms $E[X_k^k]$, $E[X_k^k X_k^k]$ and $E[X_k^k X_k^i]$ $k=1,...,N$ (see [21]). To obtain these values one may apply the procedure described above to derive $E[X_k^k]$ and $E[X_k^k X_k^k]$, $k=1, \cdots, N$. An application of a similar procedure can provide the value of $E[X_k^k X_k^i]$, and N such applications will provide all terms $E[X_k^k X_k^i]$, $k=1, \cdots, N$. Therefore, the derivation of $E[W_i]$ will require $O(N^2 \log_a \varepsilon)$ steps where a is the convergence factor given in [21].

F. Second and Higher Moments of Delay

As observed by Eisenberg [11], the derivation of $E[W_i^2]$ (second moment of delay) can be done by deriving the first three moments of X_i^i (buffer occupancy). Ferguson [26] suggested the station time method to derive the second moments of W_i. The application of the station time approach requires the derivation and the use of a relatively complex set of linear equations whose size is N^3. The amount of computation required by the method seems[2] to be

[2] Complexity analysis of the method is not provided in the literature, and thus, in this framework we can only conjecture it.

at least $O(N^3 \log_\rho \varepsilon)$ steps.

As we show next, by using *very simple* equations, the descendant set approach can be easily adapted to allow the derivation of $E[(X_1^1)^3]$ and thus of delay second moments. Moreover, since this adapted procedure differs from the one described in Section III by only very few equations, its overall computation complexity *remains* $O(N \log_\rho \varepsilon)$ steps per station. This forms a very significant reduction in computation complexity in comparison with the previously known methods.

To demonstrate the application of the descendant set approach to second moment derivation we apply it to the cyclic system with gated service; the extension to the other variations presented in this paper follows. As stated earlier, the key for deriving $E[(W_1)^2]$ is the derivation of $E[(X_1^1)^3]$ (in addition to the first and second moments whose derivation is provided in Section III.C, III.D and III.E). To this end, it is easy to consider Equations 3.1a-b and to derive from them third moment recursive relationship (similar to the second moment relationship of (3.3)). This yields,

$$\alpha_{i,c}^{(3)} = b_i^{(3)} \cdot \left[\left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}\right]^3\right] + b_i \cdot \left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c}^{(3)} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}^{(3)}\right] + 3 b_i^{(2)} \cdot \left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c}^{(2)} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}^{(2)}\right] \cdot \left[\left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}\right]^2\right] \quad c \geq 0 \tag{4.4}$$

In a similar manner we may now consider Equations (3.4a-b) and derive from them the proper third moment recursive relationship (similar to (3.5)). This yields:

$$\beta_{i,c}^{(3)} - \beta_{i,c}^3 = (r_i^{(3)} - r_i^3) \cdot \left[\left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}\right]^3\right] + r_i \cdot \left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c}^{(3)} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}^{(3)}\right] + 3 r_i^{(2)} \cdot \left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c}^{(2)} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}^{(2)}\right] \cdot \left[\left[\sum_{j=i+1}^{N} \lambda_j \alpha_{j,c} + \sum_{j=1}^{i} \lambda_j \alpha_{j,c-1}\right]^2\right] \quad c \geq 0 \tag{4.5}$$

Finally, we may use the relation $X_1^1 = \sum_{c=0}^{\infty} \sum_{i=1}^{N} R_{i,c}$ to derive an expression for $E[(X_1^1)^3]$ (similar to (3.9b)), yielding:

$$E[(X_1^1)^3] = E\left[\left[\sum_{c=0}^{\infty} \sum_{i=1}^{N} R_{i,c}\right]^3\right] = 3\left[\sum_{c=0}^{\infty} \sum_{i=1}^{N} \beta_{i,c}^{(2)} - \beta_{i,c}^2\right]\left[\sum_{c=0}^{\infty} \sum_{i=1}^{N} \beta_{i,c}\right] + \left[\sum_{c=0}^{\infty} \sum_{i=1}^{N} \beta_{i,c}\right]^3 + \sum_{c=0}^{\infty} \sum_{i=1}^{N} \beta_{i,c}^{(3)} - \beta_{i,c}^3 \tag{4.6}$$

The modification of the algorithm described in Section III.E is now rather simple. This is done by adding an array alpha_3[1...*N*] (similar to alpha_2) and variables sum_alpha_3 and sum_beta_3 (similar to sum_alpha_2 and sum_beta_2, respectively). The update of these variables is done similarly to the updates provided in the algorithm before, and using relations (4.4), (4.5). Note that the only change in the algorithm is the addition of several assignments in the inner iteration and thus the order of magnitude of the number of operations it requires does not change and remains $O(N \log_\rho \varepsilon)$.

It should be pointed out that in a similar manner higher moments of X_1^1 and of W_1 can be derived as well. It is clear that the number of operations required by such analysis remains $O(N \log_\rho \varepsilon)$.

G. Mixed Feature Systems

In Sections IV.A - IV.F we described various variations of the basic system which can be handled by the descendant set approach. Obviously, any reasonable combination of these models can be analyzed by the method. For instance, using a combination of the equations derived in this section one can use the method to derive the second moment of the delay in a system with binomial-gated service policy, an arbitrary periodic order of service (polling table) and customer routing.

V. Discussion and Concluding Remarks

The descendant-set computation approach is an efficient method for the computation of delay moments in polling systems. The number of operations required by the method, per iteration, when applied to a single station is given in the first row of Table 1. The number of iterations required by the method is logarithmic in the accuracy and depends on the system parameters.

To evaluate the merits of the method we next compare it to the algorithms which are most commonly used for this analysis. The number of operations required by the station time approach (per iteration) for the computation of all N delay figures is given in the second row of Table 1. Note that in the table one entry is marked by "-", denoting that the method is not applicable (or has not been applied) to the corresponding feature. Also, the number of operations required for the derivation of the kth moment is conjectured here on the basis of previous research (such derivation has not been carried out). The convergence factor for each of the features is identical to that of the descendant-set approach. The number of operations required by Sarkar and Zangwill's approach for the computation of all N delay figures is given in the third row of Table 1.

The comparison shows that the descendant-set is superior to the other methods in most ranges of parameters. This superiority is mostly expressed when not all N delay figures are required (this is the case either when the analysis of

TABLE 1: COMPARISON OF COMPUTATION METHODS

FEATURE	Basic System	Polling Table	Customer Routing	Second Moment	kth Moment
Descendant set: number of operations per iteration (single station computation)	$O(N)$	$O(M)$	$O(N^2)$	$O(N)$	$O(N)$
Station time: number of operations per iteration (N stations computation)	$O(N^2)$	$O(M^2)$	–	$O(N^3)$	$O(N^{k+1})$
Sarkar & Zangwill: number of operations (N stations computation)	$O(N^3)$	–	–	–	–

only a few stations is required or when many stations are similar to each other). Very strong superiority is expressed also for high moments of delay.

Additional advantage of the descendant set method, in comparison to the other methods, is that it applies to a wider set of polling systems models and variations. The only method that applies to this wide set of variations is the buffer occupancy method (whose computational complexity is, nontheless, much higher). In fact, since the sets of variables used by both methods are identical (buffer occupancy variables) we believe that a fixed visit-order model which will be analyzable by the buffer occupancy approach will also be analyzable by the descendant set approach. For example, such a model is a system with correlated arrivals which has been analyzed in Levy and Sidi [27] using the buffer occupancy approach, and can be analyzed by the descendant set approach as well.

Another advantage of the method is in the simple structure of its equations (which allows the separation of one station analysis from that of the others). The simplicity of the structure may be useful for better understanding of these systems and in optimization studies.

REFERENCES

[1] O. J. Boxma, H. Levy and J. A. Weststrate, "Optimization of Polling Systems," in: *Performance '90*, eds. P.J.B King, I. Mitrani and R.J. Pooley (North-Holland, Amsterdam), pp. 349-361, Sept. 1990.

[2] O.J. Boxma, H. Levy and J.A. Weststrate, "Efficient Visit Frequencies for Polling Tables: Minimization of Waiting Cost" *Queueing Systems and their Applications*, vol. 9, pp. 133-162, 1991.

[3] D. Sarkar and W. I. Zangwill, "Expected Waiting Time for Nonsymmetric Cyclic Queueing Systems - Exact Results and Applications," *Management Science*, vol. 35, pp. 1463-1474, Dec. 1989.

[4] G. B. Swartz, "Polling in a loop system," *J. Ass. Comp. Mach.*, vol. 27, pp. 42-59, Dec. 1980.

[5] L. D. Servi, "Average Delay Approximation of M/G/1 Cyclic Service Queues with Bernoulli Schedules," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-4, pp. 813-822, Sept. 1986.

[6] K.K. Leung, "Waiting time distributions for token-passing systems with limited-K service via discrete Fourier transforms, in: *Performance '90*, eds. P.J.B. King, I. Mitrani, R.J. Pooley (North-Holland, Amsterdam), pp. 333-347, Sept. 1990.

[7] K.K. Leung and M. Eisenberg, "A single-server queue with vacations and gated time-limited service," *IEEE Trans. on Commun.*, vol. 38, pp. 1454-1462, Sept. 1990.

[8] Tedijanto, "Exact Analysis for the Cyclic-Service Queue with a Bernoulli Schedule," *Performance Evaluation*, vol. 11, pp. 107-115, 1990.

[9] R.O. LaMaire, "An M/G/1 vacation model of an FDDI station," *IEEE J. on Sel. Areas in Commun.*, vol. 9, pp. 257-264, Feb. 1991.

[10] H. Takagi, "*Analysis of Polling Systems*," MIT Press, Apr. 1986.

[11] M. Eisenberg, "Queues with Periodic Service and Changeover Time," *Operations Research*, vol. 20, pp. 440-451, Mar.-Apr. 1972.

[12] J.E. Baker and I. Rubin, "Polling with a General-Service Order Table," *IEEE Trans. on Commun.*, vol. COM-35, pp. 283-288, Mar. 1987.

[13] G.L. Choudhury, "Polling With A General Service Order Table : Gated Service," Proceedings of the *IEEE INFOCOM '90*, San Francisco, CA, USA, pp. 268-276, June 1990.

[14] L. Kleinrock and H. Levy, "The Analysis of Random Polling Systems," *Operations Research*, vol. 36, pp. 716-332, Sept.-Oct. 1988.

[15] O.J. Boxma, J.A., Weststrate, Waiting times in polling systems with Markovian server routing. *In: G. Stiege and J.S. Lie (eds.), Messung, Modellierung und Bewertung von Rechensystemen und Netzen (Springer, Berlin), pp. 89-104, Sept. 1989.*

[16] M.M. Srinivasan, "Nondeterministic Polling Systems," *Management Science*, vol. 37, pp. 667-681, June 1991.

[17] H. Levy, "Binomial-Gated Service: A Method for Effective Operation and Optimization of Polling Systems" *IEEE Trans. on Commun.*, vol. 39, pp. 1341-1350, Sep. 1991.

[18] O.J. Boxma, "Workloads and waiting times in single-server systems," *Queueing Systems and Their Applications*, vol. 5, pp. 185-214, 1989.

[19] S. Browne and U. Yechiali, "Dynamic server routing in binomial-gated, binomial-exhaustive, Bernoulli-gated, Bernoulli-exhaustive and mixed polling systems," Technical Report, Department of Statistics, Tel-Aviv University, 1989.

[20] H. Levy, "Optimization of Polling Systems: The Fractional-Exhaustive Service Method," Report, Department of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, Nov. 1988.

[21] M. Sidi, H. Levy and S.W. Fuhrmann, "A Queueing Network with a Single Cyclically Roving Server," *Queueing Systems and their Applications*, vol. 11, pp. 121-144, 1992.

[22] S.W. Fuhrmann and R.B. Cooper, "Stochastic decompositions in the M/G/1 queue with server vacations," *Operations Research*, vol. 32, pp. 1368-1373, Sept.-Oct. 1985.

[23] O.J. Boxma and W.P. Groenendijk, "Pseudo conservation laws in cyclic service systems," *Journal of Applied Probability*, vol. 24, pp. 949-964, Dec. 1987.

[24] E.G. Coffman and A. Stoylar, "Continuous Polling on Graphs," *Prob. Eng. and Info. Sc.*, vol. 7, no. 2, pp. 209-226, 1993.

[25] H. Levy, "Delay Computation and Dynamic Behavior of Polling Systems," *Performance Evaluation*, vol. 10, pp. 35-51, Oct. 1989.

[26] M.J. Ferguson, "Computation of the Variance of the Waiting Time for Token Rings," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-4, pp. 775-782, Sept. 1986.

[27] H. Levy and M. Sidi, "Polling Systems with Simultaneous Arrivals," *IEEE Trans. on Commun.*, vol. 39, pp. 823-827, June 1991.

Alan G. Konheim received the B.E.E. and M.S. (Mathematics) degree from the Polytechnic Institute of Brooklyn in 1955 and 1957 and the Ph.D. (Mathematics) degree from Cornell University in 1960. From 1960-82 he was a Research Staff Member of the Mathematics Sciences Department at IBM Research. He joined the faculty of the Computer Science Department in 1982. His interests are in computer communication and cryptography.

Hanoch Levy received the B.A degree in Computer Science with distinctions from the Technion, Israel Institute of Technology in 1980 and the M.Sc and the Ph.D degrees in Computer Science from University of California at Los Angeles, in 1982 and 1984, respectively.

From 1984 to 1987 he was a member of technical staff in the Department of Teletraffic Theory at AT&T Bell Laboratories. In 1987 he joined the Department of Computer Science, Tel-Aviv University, Tel-Aviv, Israel. Currently he is at RUTCOR, Rutgers University, New-Brunswick, NJ, on leave from Tel-Aviv University. His interests include Computer Communication Networks, Performance Evaluation of Computer Systems and Queueing Theory.

Mandyam M. Srinivasan is an Associate Professor in the Management Science Program at the University of Tennessee, Knoxville. He received the Master of Technology degree from the Indian Institute of Technology, Madras, India, a Post-Graduate Diploma in Management (M.B.A.) from the Indian Institute of Management, Bangalore, India, and a Ph.D. in Industrial Engineering and Management Sciences from Northwestern University.

Dr. Srinivasan's current research interests are in performance evaluation of computer networks and flexible manufacturing systems. He is a member of The Institute of Management Sciences and the Operations Research Society of America. He is an Associate Editor of International Journal of Flexible Manufacturing Systems, and serves on the Editorial Review Board of Production and Operations Management.