JOURNAL OF AIRCRAFT
Vol. 36, No. 5, September–October 1999

Mean Value Analysis of Military Airlift Operations at an Individual Airfield

Dennis C. Dietz*
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433

An analytical queueing network model of military airlift operations at an individual airfield is developed and implemented. The airfield system is represented as a capacitated open network of multiserver queues and fork-join constructs that capture concurrent aircraft flow through subnetworks of activities. Generated performance measures include mean values for system throughput, aircraft ground time, and resource utilizations. The quality of the analytical approximation is demonstrated through comparison with simulation results.

I. Introduction

ONE of the primary missions of the U.S. Air Force is air mobility, defined as the rapid, global airlift of combat forces and supporting units. Planning and execution of this complex mission has been systematically improved through the exercise of large-scale simulation and network optimization models. In these system-level models, operations at individual airfields are generally represented by a few aggregate parameters such as average aircraft ground time or throughput.[1] These parameters can be difficult to determine because they are affected by interrelated service activities performed on aircraft transiting the airfield. Some of these activities require utilization of constrained resources. Because of the nonlinear interactions between relevant factors, detailed airfield operations have historically been studied through simulation methods. However, substantial insight and reasonable parameter estimates can be rapidly obtained through the analytical model developed in this paper.

II. Model Description

As suggested by the example model depicted in Fig. 1, each aircraft landing at an airfield can be viewed as a customer entering a capacitated open queueing network. The capacity N is simply the maximum number of aircraft that can simultaneously occupy the airfield due to some limiting factor, e.g., available parking positions. As customer population approaches airfield capacity, a command and control system must intervene to divert additional arrivals. Arriving aircraft proceed through a network of activities including landing, parking, cargo handling, refueling, liquid oxygen (LOX) servicing, taxi-out, and takeoff. Two types of scheduled maintenance are also captured in the network model: maintenance that can be accomplished concurrently with refueling and LOX servicing (C Mx), and maintenance that cannot be accomplished concurrently with these activities (N Mx). Because some activities require the utilization of constrained resources, queueing delay may occur at some network stations. For example, the takeoff activity may involve some waiting time for a single resource (the runway). In the example network, no queueing occurs for taxiways or maintenance resources. The corresponding network stations are thus modeled as "infinite server" or delay stations. For simulation purposes, service times are assigned deterministic (D) values for parking and taxi activities, and are modeled with Erlang-2 (E_2) random variables for all other activities. While the analytical model developed in this paper relies on exponentially distributed service times for all network stations, a generalized simulation is employed to verify the accuracy of the model and illustrate the relative insensitivity of results to differences in higher-order moments. Mean service time for each activity i is denoted by s_i (in hours).

Another important feature of the queueing network formulation is the inclusion of "fork-join" constructs (represented in Fig. 1 with diamond-shaped symbols). Within these constructs, multiple activities can be performed concurrently if the required resources are available. Thus, aircaft arriving at a fork node can be viewed as generating temporary clones that are rejoined into a single customer at the corresponding join node when all activities along each clone path are complete. The probability that path k is required for an aircraft arriving at fork node j is given as q_{jk}. For the example network, all paths are always required except path $B2$ (refueling and LOX servicing), which is required with probability $q_{B2} = 0.47$.

To develop an efficient solution algorithm, it is helpful to transform the open capacitated queueing network into an equivalent closed network. This transformation is accomplished by inserting a new single-server station into the network, and artificially setting the population of the closed network equal to the original network capacity (see Fig. 2). The new arrival station (labeled 0), has an exponentially distributed service time and essentially represents the portion of the entire airlift system that operates outside the airfield boundary. When all N customers occupy the airfield portion of the network (stations 1–9), station 0 is idle and no further arrivals can be generated. However, when the airfield is not occupied at capacity, at least one customer occupies station 0, and so arrivals are generated according the Poisson arrival rate λ.

The network transformation permits the calculation of several interesting performance measures. First, assume an algorithm exists to compute local performance measures for each station i. These measures include mean response time R_i (waiting and service time), throughput λ_i, queue length Q_i (number of customers waiting or in service), and server utilization U_i (expected number of busy servers). Network performance measures can then be derived from the local results for station 0. For example, because λ_0 is the customer throughput for station 0, it follows that N/λ_0 is the mean cycle time for a particular customer. A mean value for airfield transit time can therefore be computed as $T = N/\lambda_0 - R_0$. Using similar reasoning, the average number of aircraft on the ground can be calculated as $AOG = N - Q_0$.

Another important performance measure is the probability that the airfield is at capacity, or the probability of airfield saturation. This value essentially represents the portion of planned arrivals that must be diverted, and is therefore a measure of required command and control intervention. Because the network is occupied at capacity only when station 0 is idle, the probability of airfield saturation can be calculated as $P_N = 1 - U_0$.

III. Mean Value Analysis

If the fork-join constructs were removed from the example network, local performance measures could be computed with the use

Received 6 July 1998; revision received 12 November 1998; accepted for publication 17 December 1998. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Adjunct Associate Professor of Operations Research, Department of Operational Sciences, 2950 P Street; currently Operations Research Consultant, U.S. WEST Advanced Technologies, 4001 Discovery Drive, Boulder, CO 80303.

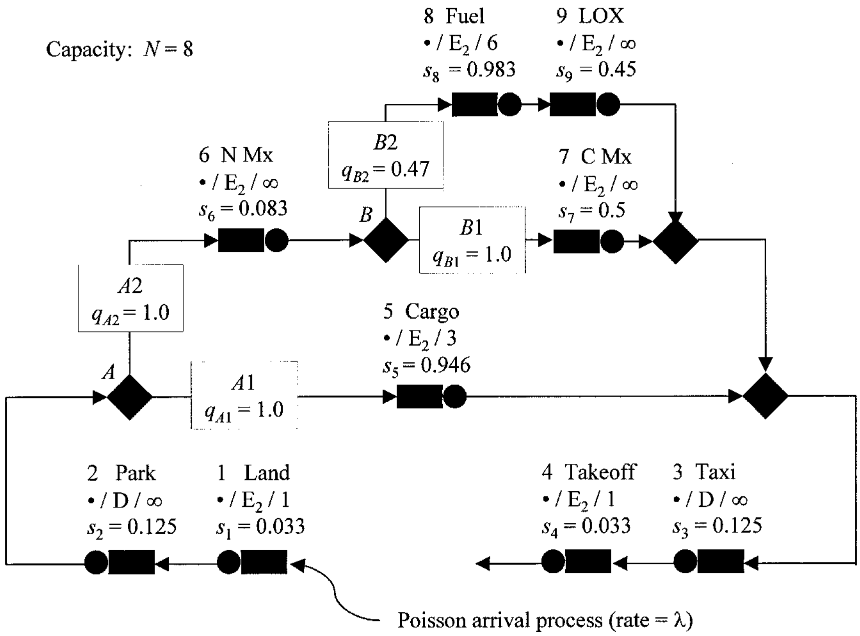

Fig. 1 Capacitated open queueing network (♦ = "fork-join" constructs).

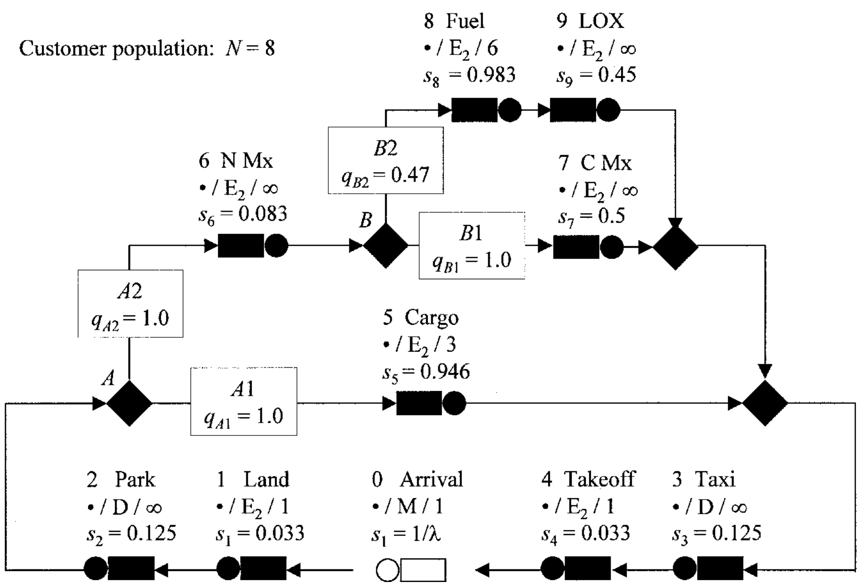

Fig. 2 Equivalent closed queueing network.

of mean value analysis (MVA). Excellent expositions of the MVA algorithm are given in Bruell and Balbo[2] or Conway and Georganas.[3] In general, MVA can be applied to a closed network with N customers circulating through M stations having symmetric service disciplines (first-come first-served exponential, processor sharing, infinite server, or last-come first-served pre-emptive).[4] Each station may consist of a single server, multiple identical servers, or an infinite number of servers. The service rate when n customers are at station i is given by $\mu_i(n) = \min(n/s, r_i/s_i)$; where r_i is the resource availability (maximum number of aircraft that can be simultaneously serviced).

A critical foundation for MVA is the arrival theorem, proven in Lavenberg and Reiser.[5] This theorem states that, for a closed network with N customers, an arriving customer observes the same distribution of customers at a station as the stationary (random observer's) distribution for the same network with $N - 1$ customers. The arrival theorem leads to the marginal local balance theorem, which states that

$$\mu_i(n)P_i(n \mid N) = \lambda_i(N)P_i(n-1 \mid N-1) \tag{1}$$

where $P_i(n \mid N)$ is the probability that n customers are at station i, given N customers are in the network, and $\lambda_i(N)$ is the customer throughput for station i when N customers are in the network. The marginal local balance theorem can be applied to recursively compute the performance measures. First, note that the mean queue length at any station i can be written as

$$Q_i(N) = \sum_{n=1}^{N} nP_i(n \mid N) = \sum_{i=1}^{N} \frac{n\lambda_i(N)}{\mu_i(n)} P_i(n-1 \mid N-1) \tag{2}$$

The throughput in Eq. (2) is unknown, but application of Little's law yields

$$R_i(N) = \sum_{n=1}^{N} \frac{n}{\mu_i(n)} P_i(n-1 \mid N-1) \tag{3}$$

Note that if station i has only one server, then

$$Q_i(N) = s_i \sum_{n=1}^{N} n P_i(n-1 \mid N-1) = s_i\{1 + Q_i(N-1)\} \tag{4}$$

and if it has an infinite number of servers, then $R_i(N) = s_i$ for all N.

Equations (3) and (4) relate the response time of a station when N customers are present in the network to the distribution of customers at the station when $N-1$ customers are present. Thus, by starting at $N = 1$, [where $P_i(0 \mid N-1) = 1$ and $Q_i(N-1) = 0$ for all i], performance measures can be obtained recursively. To determine station throughputs at each iteration, it is necessary to determine the average cycle time for a customer at an arbitrary reference station, e.g., station 0. The average time between two successive departures by the same customer from station 0 is given by

$$CT_0(N) = \sum_{i=0}^{M-1} \frac{v_i R_i}{v_0} \tag{5}$$

where each ratio v_i/v_0 is the mean number of visits a customer makes to station i for every visit to station 0. Using the cycle time from Eq. (5), throughput at each station can be determined as

$$\lambda_i(N) = \frac{N v_i}{CT_0(N) v_0} \tag{6}$$

By applying Little's law, queue length and utilization can then be computed as

$$Q_i(N) = R_i(N)\lambda_i(N) \tag{7}$$

$$U_i(N) = s_i \lambda_i(N) \tag{8}$$

If station i has a single server, then the result from Eq. (6) can be used in Eq. (4) to obtain response times with more than one customer in the network. If station i has multiple servers, then the marginal local balance theorem can be applied to determine the new distribution of customers as

$$P_i(n \mid N) = \frac{\lambda_i(N) P_i(n-1 \mid N-1)}{\mu_i(n)}, \qquad n > 0 \tag{9}$$

$$P_i(0 \mid N) = 1 - \sum_{n=1}^{N} P_i(n \mid N) \tag{10}$$

These probabilities can then be used in Eq. (3) to obtain the multiserver response times for the next iteration. The process is repeated until N is equal to the desired number of customers in the network.

IV. Fork-Join Constructs

With the addition of concurrent activity paths, the product-form nature of the network is destroyed and the MVA algorithm can no longer be directly applied. However, a heuristic based on MVA was recently developed by Rao and Suri[6] to analyze a single fork-join system of single-server queues in a manufacturing context. Dietz and Jenkins[7] subsequently extended the heuristic to accommodate multiserver stations, probabilistic service requirements, and multiple fork-join constructs in a sortie generation network for fighter aircraft. This paper further extends the heuristic approach to address the possibility of multiple stations on a fork-join path, including nested fork-join constructs.

Consider the case of a fork-join construct j containing K_j paths, where $q_{jk} = 1$ for all $k = 1, \ldots, K_j$. A customer arriving at fork node j immediately generates K_j clones, each of which enters the queue for the first station on its respective path. The parent customer can be viewed as holding position at fork node j until all of its clones have transited their paths, at which time the parent instantaneously moves to the join node and proceeds through the network. Mean response time, queue length, and utilization can be estimated for clone traffic at each station on each path provided two key approximations are adopted:

1) Approximation 1: For a network with N customers, a clone arriving at a station sees the stationary (random observer's) distribution of clones at the station for the same network with $N-1$ customers.

2) Approximation 2: The transit time of a clone along a fork-join path can be represented as an exponentially distributed random variable and is independent of the transit time for clones on other paths.

Based on approximation 2, the transit time for any path jk can be denoted by an exponential random variable $T_{jk}(N)$ with rate parameter $\theta_{jk}(N) = 1/E[T_{jk}(N)]$. The mean time that a parent customer holds at fork node j, which must be determined to obtain network cycle time, is $E[\max_{k=1,\ldots,K_j}\{T_{jk}(N)\}]$.

The assumption that $q_{jk} = 1$ can be relaxed by simply conditioning on the subset of service activities S that may be required by a customer. Let Ω_i be the union of all possible subsets for a particular fork-join node j, and let $\pi_j(S)$ be the probability that subset S is required. The number of subsets in S is given by

$$\sum_{k=0}^{K_j} \binom{K_j}{k} = 2^{K_j} \tag{11}$$

If suitable system data were available, exhaustive probabilities $\pi_j(S)$ could be explicitly assigned to each subset. Under the assumption that all fork-join paths are required independently, the probability that a customer requires a particular subset S at fork-join node j is given by

$$\pi_j(S) = \prod_{k \in S} q_{jk} \prod_{k \notin S} (1 - q_{jk}) \tag{12}$$

The mean of the associated conditional holding time at the fork node is $E[\max_{k \in S}\{T_{jk}(N)\}]$ (defined to be zero if $S = \emptyset$). The computation of mean holding time mirrors a familiar problem from reliability theory, i.e., determining the mean time to failure for a parallel system of independent components with exponentially distributed failure times.[8] Because the path transit times are assumed to be independent by approximation 2, the cumulative distribution function (CDF) for conditional holding time is

$$\begin{aligned} F(t) &= \prod_{k \in S} P\{T_{jk}(N) \le t\} \\ &= \prod_{k \in S} (1 - \exp\{-\theta_{jk}(N)t\}) \end{aligned} \tag{13}$$

It is well known that

$$E[X] = \int_0^\infty [1 - F(t)]\, \mathrm{d}t$$

for any nonnegative continuous random variable X with CDF $F(t)$, so

$$\begin{aligned} E\left[\max_{k \in S}(T_{jk})\right] &= \int_0^\infty \left\{1 - \prod_{k \in S}(1 - \exp\{-\theta_{jk}(N)t\})\right\} \mathrm{d}t \\ &= \int_0^\infty \Bigg\{1 - 1 + \sum_{k \in S} \exp\{-\theta_{jk}(N)t\} \\ &\quad - \sum_{k \in S} \sum_{\substack{l \in S \\ l < k}} \exp\{-(\theta_{jk}(N) + \theta_{jl}(N))t\} \\ &\quad + \sum_{k \in S} \sum_{\substack{l \in S \\ l < k}} \sum_{\substack{m \in S \\ m < l}} \exp\{-(\theta_{jk}(N) + \theta_{jl}(N) + \theta_{jm}(N))t\} \\ &\quad - \cdots + (-1)^{K(S)+1} \exp\left\{-\left(\sum_{k \in S} \theta_{jk}(N)\right)t\right\}\Bigg\} \mathrm{d}t \end{aligned} \tag{14}$$

where $K(S)$ is the number of paths in S. Evaluation of the integral yields

$$E\Big[\max_{k\in S}\{T_{jk}\}\Big]=\sum_{k\in S}\frac{1}{\theta_{jk}(N)}-\sum_{k\in S}\sum_{\substack{l\in S\\ l<k}}\frac{1}{\theta_{jk}(N)+\theta_{jl}(N)}$$

$$+\sum_{k\in S}\sum_{\substack{l\in S\\ l<k}}\sum_{\substack{m\in S\\ m<l}}\frac{1}{\theta_{jk}(N)+\theta_{jl}(N)+\theta_{jm}(N)}$$

$$-\cdots+(-1)^{K(S)+1}\frac{1}{\sum_{k\in S}\theta_{jk}(N)} \tag{15}$$

The MVA algorithm can now be modified to accommodate a general network composed of simple service stations and fork-join constructs. At each iteration, computation of cycle time is modified by applying the fork-join approximations and conditioning on the sets of fork-join paths taken. Referring to the example network in Fig. 2, we first note that

$$E[T_{B1}(N)]=R_7(N) \tag{16}$$

$$E[T_{B2}(N)]=R_8(N)+R_9(N) \tag{17}$$

By approximation 2 and Eq. (15), mean holding time at fork node B is given by

$$E[T_B(N)]=\pi_B(\{1,2\})E[\max\{T_{B1}(N),\ T_{B2}(N)\}]$$

$$+\pi_B(\{1\})E[T_{B1}(N)]\approx(0.47)\left\{\frac{1}{\theta_{B1}(N)}+\frac{1}{\theta_{B2}(N)}\right.$$

$$\left.-\frac{1}{\theta_{B1}(N)+\theta_{B2}(N)}\right\}+(0.53)\left\{\frac{1}{\theta_{B1}(N)}\right\} \tag{18}$$

Similarly, holding time for fork node A is determined as

$$E[T_{A1}(N)]=R_5(N) \tag{19}$$

$$E[T_{A2}(N)]=R_6(N)+T_B(N) \tag{20}$$

and so, by approximation 2 and Eq. (15)

$$E[T_A(N)]=\pi_A(\{1,2\})E\{\max[T_{A1}(N),\ T_{A2}(N)]\}$$

$$\approx(1)\left\{\frac{1}{\theta_{A1}(N)}+\frac{1}{\theta_{A2}(N)}-\frac{1}{\theta_{A1}(N)+\theta_{A2}(N)}\right\} \tag{21}$$

Table 1 Performance results for the example model

			Queue length, Q_i			Utilization, U_i		
Activity	R_i	λ_i	MVA	Simulation	% error	MVA	Simulation	% error
				$\lambda = 1.0$				
1 Land	0.034	1.000	0.034	0.034 ± 0.000	+0.838	0.033	0.033 ± 0.000	——
2 Park	0.125	1.000	0.125	0.125 ± 0.000	——	0.125	0.125 ± 0.000	——
3 Taxi	0.125	1.000	0.125	0.125 ± 0.000	——	0.125	0.125 ± 0.000	——
4 Takeoff	0.034	1.000	0.034	0.034 ± 0.000	+0.790	0.033	0.033 ± 0.000	——
5 Cargo	0.974	1.000	0.980	0.974 ± 0.003	+0.575	0.945	0.946 ± 0.003	——
6 N Mx	0.083	1.000	0.083	0.083 ± 0.000	——	0.083	0.083 ± 0.000	——
7 C Mx	0.500	1.000	0.500	0.500 ± 0.001	——	0.500	0.500 ± 0.001	——
8 Fuel	0.984	0.470	0.462	0.463 ± 0.002	——	0.462	0.463 ± 0.002	——
9 LOX	0.449	0.470	0.211	0.211 ± 0.001	——	0.211	0.211 ± 0.001	——
0 Arrival	6.298	1.000	6.135	6.296 ± 0.004	−2.556	1.000	1.000 ± 0.000	−0.022
				$\lambda = 2.0$				
1 Land	0.035	1.922	0.068	0.068 ± 0.000	——	0.063	0.064 ± 0.000	−1.489
2 Park	0.125	1.922	0.240	0.244 ± 0.000	−1.526	0.240	0.244 ± 0.000	−1.526
3 Taxi	0.125	1.922	0.240	0.244 ± 0.000	−1.524	0.240	0.244 ± 0.000	−1.524
4 Takeoff	0.034	1.922	0.068	0.067 ± 0.000	+0.486	0.063	0.064 ± 0.000	−1.480
5 Cargo	1.133	1.922	2.179	2.194 ± 0.006	−0.697	1.818	1.847 ± 0.004	−1.567
6 N Mx	0.083	1.922	0.160	0.162 ± 0.000	−1.565	0.160	0.162 ± 0.000	−1.565
7 C Mx	0.500	1.922	0.961	0.976 ± 0.001	−1.486	0.961	0.976 ± 0.001	−1.486
8 Fuel	0.983	0.903	0.888	0.901 ± 0.002	−1.436	0.888	0.901 ± 0.002	−1.434
9 LOX	0.450	0.904	0.407	0.413 ± 0.001	−1.560	0.407	0.413 ± 0.001	−1.597
0 Arrival	2.179	1.922	4.189	4.482 ± 0.007	−6.530	0.961	0.977 ± 0.000	−1.576
				$\lambda = 3.0$				
1 Land	0.036	2.532	0.091	0.093 ± 0.000	−2.127	0.084	0.087 ± 0.000	−3.820
2 Park	0.125	2.532	0.317	0.329 ± 0.000	−3.879	0.317	0.329 ± 0.000	−3.879
3 Taxi	0.125	2.532	0.317	0.329 ± 0.000	−3.877	0.317	0.329 ± 0.000	−3.877
4 Takeoff	0.034	2.532	0.091	0.092 ± 0.000	−1.340	0.084	0.087 ± 0.000	−3.821
5 Cargo	0.036	2.532	3.307	3.572 ± 0.008	−7.856	2.396	2.492 ± 0.004	−3.856
6 N Mx	0.083	2.532	0.210	0.219 ± 0.000	−3.850	0.210	0.219 ± 0.000	−3.850
7 C Mx	0.500	2.532	1.266	1.317 ± 0.002	−3.832	1.266	1.317 ± 0.002	−3.832
8 Fuel	0.983	1.190	1.170	1.214 ± 0.004	−3.610	1.170	1.214 ± 0.004	−3.615
9 LOX	0.450	1.190	0.536	0.557 ± 0.001	−3.851	0.536	0.557 ± 0.001	−3.851
0 Arrival	1.041	2.532	2.636	2.830 ± 0.008	−6.830	0.843	0.877 ± 0.000	−3.838
				$\lambda = 4.0$				
1 Land	0.036	2.832	0.102	0.104 ± 0.000	−1.794	0.094	0.097 ± 0.000	−3.906
2 Park	0.125	2.832	0.354	0.369 ± 0.000	−4.038	0.354	0.369 ± 0.000	−4.038
3 Taxi	0.125	2.832	0.354	0.369 ± 0.000	−4.036	0.354	0.369 ± 0.000	−4.036
4 Takeoff	0.036	2.832	0.102	0.103 ± 0.000	−0.977	0.094	0.097 ± 0.000	−3.982
5 Cargo	1.426	2.832	4.038	4.594 ± 0.008	−12.100	2.679	2.791 ± 0.002	−4.005
6 N Mx	0.083	2.832	0.235	0.245 ± 0.000	−4.034	0.235	0.245 ± 0.000	−4.034
7 C Mx	0.500	2.832	1.416	1.475 ± 0.002	−4.021	1.416	1.475 ± 0.002	−4.021
8 Fuel	0.983	1.331	1.309	1.362 ± 0.004	−3.941	1.309	1.362 ± 0.004	−3.944
9 LOX	0.440	1.331	0.599	0.623 ± 0.002	−3.904	0.599	0.623 ± 0.002	−3.904
0 Arrival	0.610	2.832	1.726	1.782 ± 0.005	−3.105	0.708	0.738 ± 0.000	−4.060

By approximation 1, cycle time can be computed within the MVA algorithm for any N as

$$
\begin{aligned}
CT_0(N) &\approx \sum_{i=0}^{4} \frac{v_i}{v_0} R_i(N) + \frac{v_A}{v_0} T_A(N) \\
&= \sum_{i=0}^{4} R_i(N) + T_A(N) \qquad (22)
\end{aligned}
$$

V. Implementation and Results

The algorithm described in the previous section has been implemented on a Pentium personal computer with a PASCAL software development environment. Run time for the example network is essentially zero (less then 1 s for all cases examined). Table 1 displays a complete set of performance results for each network station with airfield capacity $N = 8$ and four different arrival rates ($\lambda \in \{1.0, 2.0, 3.0, 4.0\}$). For a realistic arrival rate of 2.0 aircraft per hour, network performance measures can be computed as follows:

1) Transit time: $T = N/\lambda_0 - R_0 = 8/1.922 - 2.179 = 1.983$ h.
2) Average on ground: $AOG = N - Q_0 = 8 - 4.189 = 3.811$.
3) Probability of airfield saturation: $P_N = 1 - U_0 = 1 - 0.961 = 0.039$.

To provide insight into model accuracy, Table 1 also displays simulation results for queue length and utilization at individual stations. Estimators for mean performance measures are shown, along with 95% confidence interval half-widths. These values were produced through batch means analysis of 10^6 hours of simulated operating time at each arrival rate (each with a 10^5 hour warm-up period). When executed on a DEC AXP (Alpha) Model 500MP computing system, the simulation analysis required 22 min of core processing time for $\lambda = 1.0$, and 68 min for $\lambda = 4.0$. For each case where an analytical performance measure falls outside the corresponding simulation confidence interval, percent error is shown. It is noteworthy that no relative error exceeds 13%, with most errors being much smaller.

Performance results related to airfield occupancy are shown in Fig. 3, including AOG and bottleneck queue length $Q_5(8)$ (queue length for cargo handling). As explained by Kant,[9] a bottleneck station eventually emerges in any closed queueing network as N is increased. A very long queue forms at this station, causing mean value analysis to eventually fail computationally as the network cycle time overflows. As $N \to \infty$, airfield throughput is limited to $\lambda_0(\infty) = (v_0/v_5)(r_5/s_5) = (1)(3/0.946) = 3.171$. As the arrival rate is increased from 1.0 to 4.0 aircraft per hour, queue length for cargo handling increases from 1 to 4 aircraft, and AOG increases from about 2 to 6 aircraft. As $\lambda \to \infty$, AOG asymptotically approaches the airfield capacity, and P_N approaches 1.0.

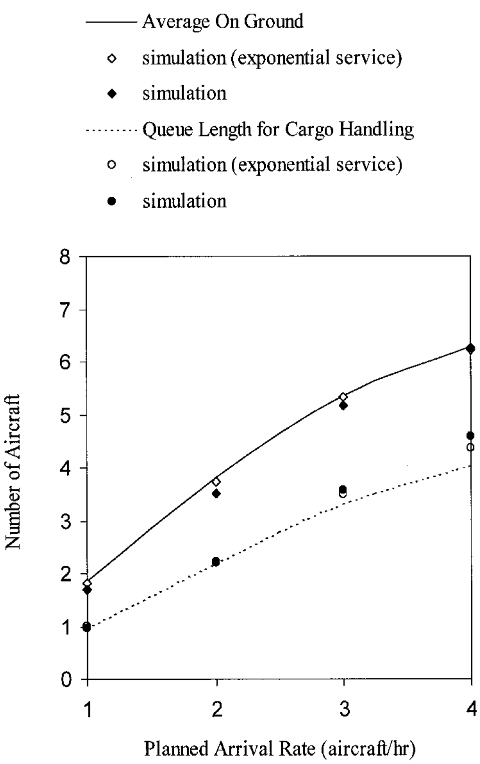

Fig. 3 Performance results for airfield occupancy.

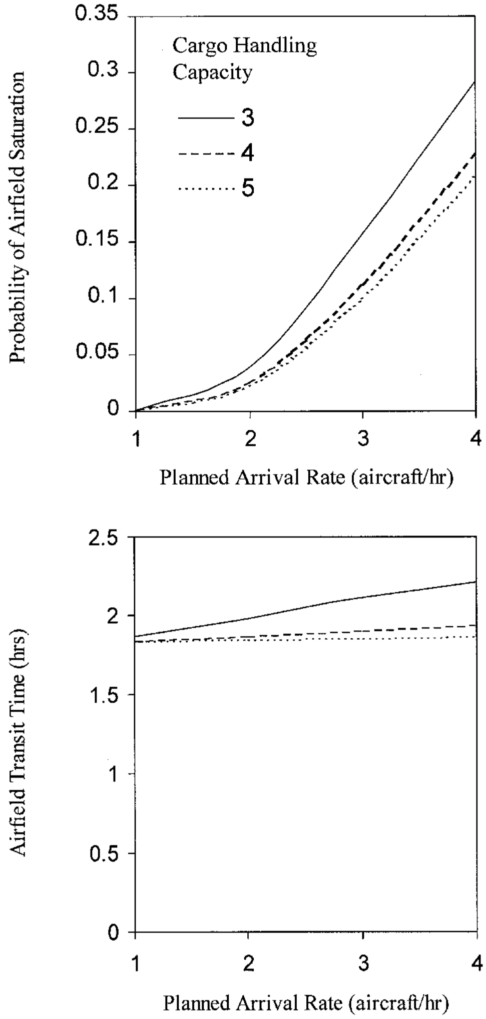

Fig. 4 Sensitivity to bottleneck capacity.

Simulation experiments were conducted using both general and exponential service times to illuminate the major sources of error. Figure 3 suggests that estimation error for AOG is due mainly to the fork-join approximations, whereas error for $Q_5(8)$ is due mainly to the exponential service time assumption. The estimation error for $Q_5(8)$ is more pronounced at high arrival rates.

Figure 4 offers some insight into the effect of improving bottleneck capacity. As resource availability for cargo handling (r_5) increases from 3 to 4 at $\lambda = 2.0$, the probability of airfield saturation decreases from 0.039 to 0.026. However, this value still increases rapidly for $\lambda > 2.0$, suggesting a practical limit on the planned arrival rate. Little marginal benefit is derived from a further increase in r_5. Airfield transit time is relatively insensitive to changes in bottleneck capacity, decreasing only by about 7% as r_5 is increased from 3 to 5. The combined results suggest that additional resources for any service activity would offer little performance improvement unless the overall capacity of the airfield can be increased.

VI. Conclusions

The analytical method presented in this paper offers reasonably accurate estimates of mean performance measures for air-mobility operations at an individual airfield. In just a few seconds, relationships can be studied that would require extensive run time with a simulation model. The method is therefore ideally suited for iterative analytical problems, such as the derivation of an optimal resource structure under specified constraints. It can also be used to quickly determine parameters for airfield representation in system-level analysis of mobility operations.

The analytical model obviously cannot capture all of the intricacies that might appear in a detailed simulation study. While the analytical results appear quite robust with respect to higher-order moments of service time distributions, steady-state operation and a constant planned arrival rate must be assumed. Nevertheless, the MVA approximation could still provide valuable insight when used as an adjunct to simulation. For example, the model could be employed to produce control variates or identify a starting point for a simulation search. It therefore should be a useful tool for anyone concerned with air-mobility analysis or comparable problems involving multiserver queues and concurrent service activities.

Acknowledgments

The author is grateful to Jean M. Steppe, Headquarters Air Mobility Command, and Craig J. Willits, U.S. Air Force Personnel Operations Agency, for their collaborative work in conceptualizing the example network and providing representative input parameters.

References

[1] Schank, J., Mattock, M., Sumner, G., Greenberg, I., Rothenberg, J., and Stucker, J., "A Review of Strategic Moblity Models and Analysis," Rept. R-3926-JS, RAND National Defense Inst., Santa Monica, CA, 1991.

[2] Bruell, S. C., and Balbo, G., *Computational Algorithms for Closed Queueing Networks*, North-Holland, New York, 1980.

[3] Conway, A. E., and Georganas, N. D., *Queueing Networks—Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*, MIT Press, Cambridge, MA, 1989.

[4] Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *Journal of the A.C.M.*, Vol. 22, No. 2, 1975, pp. 248–260.

[5] Lavenberg, S. S., and Reiser, M., "Stationary Probabilities at Arrival Instants for Closed Queueing Networks with Multiple Types of Customers," *Journal of Applied Probability*, Vol. 17, 1980, pp. 1048–1061.

[6] Rao, P. C., and Suri, R., "Approximate Queueing Network Models for Closed Fabrication/Assembly Systems. Part I: Single Level Systems," *Production and Operations Management*, Vol. 3, No. 4, 1994, pp. 244–275.

[7] Dietz, D. C., and Jenkins, R. C., "Analysis of Aircraft Sortie Generation with the Use of a Fork-Join Queueing Network Model," *Naval Research Logistics*, Vol. 44, No. 2, 1997, pp. 153–164.

[8] Kapur, K. C., and Lamberson, L. R., *Reliability in Engineering Design*, Wiley, New York, 1977.

[9] Kant, K., *Introduction to Computer System Performance Evaluation*, McGraw–Hill, New York, 1992.