JOURNAL OF AIRCRAFT
Vol. 38, No. 5, September–October 2001

Nested Fork-Join Queueing Network Model for Analysis of Airfield Operations

Craig J. Willits*
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433
and
Dennis C. Dietz†
Qwest Communications International, Boulder, Colorado 80303

This article presents a nested fork-join queueing network model of the synchronized ground processing of aircraft transiting an airfield. The queueing network is analyzed using a decomposition algorithm that provides approximate network performance measures such as throughput and expected queue lengths. The results produced are comparable in accuracy to those produced by simulation, but are generated in much less elapsed time. Using a case study of contingency operations at a military mobility airfield, we demonstrate the model's utility for rapidly developing important insights into operational performance.

I. Introduction

THE success of a modern military campaign often depends on the rapid air transport of critical resources to a distant theater of operations. To accomplish this mission, the U.S. Air Force employs a fleet of large cargo aircraft to move people and equipment through a worldwide network of airfields. To plan and execute large deployments, commanders and mobility planners rely on sophisticated modeling and analysis methods to provide meaningful estimates of operational capability. Particular attention must be paid to throughput capacity and resource bottlenecks at key airfields.

To gain insight into airfield performance, transportation analysts normally study the flow of aircraft through a series of synchronized ground processing activities using high-resolution simulation modeling techniques. In this paper, however, we approach the problem through an analytical queueing network model. Our approach is not meant to completely replace high-resolution modeling, which may be necessary for studying complex airfield processing or resource allocation schemes. Rather, the approach is designed to provide rapid insights into the relationship between airfield resource levels, the flow of mobility aircraft, and airfield throughput capacity. The model is more general than an earlier method offered by Dietz,[1] which required restrictive assumptions about service time distributions and the aircraft arrival process.

II. Analytical Airfield Model

The general task precedence graph for the ground flow of an aircraft is shown in Fig. 1. In this graph each box represents a task that may be required when an aircraft visits the airfield. The flow of precedence among tasks is from top to bottom, that is, tasks at the top of the graph must be completed before the tasks below them. Tasks that begin at the same time are immediately preceded by an appropriately labeled horizontal bar. "Refuel" and "concurrent maintenance" are examples of tasks that can begin simultaneously. If a bar labeled "synchronize" immediately follows two or more tasks, those tasks must all be complete before any task below the bar can begin. For example, the "liquid oxygen servicing" and "cargo on" tasks must be complete before the "backout/taxi" task can begin.

If we model the service providers as "stations," the aircraft as "customers" that move through the network of stations, and the number of parking spots as the network capacity N, we can construct an open capacitated queueing network[2] that is logically equivalent to the flow of tasks shown in Fig. 1. To develop an effective solution algorithm, it is helpful to transform the open queueing network into an equivalent closed network. This transformation is accomplished by inserting an "arrival" station into the network and artificially setting the population of the closed network equal to the original network capacity. The arrival station essentially represents the portion of the entire airlift system that operates outside the airfield boundary. When all N customers saturate the airfield portion of the network, the arrival station is idle, and no further arrivals can be generated. However, when the airfield is not occupied at capacity at least one customer occupies the arrival station, and so arrivals are generated at specified intervals (artificial service times for the single-server station).

The graph of the resulting closed nested fork-join queueing network, which we will call the Analytical Airfield Model (AAM), is shown in Fig. 2. The diamond symbols containing F and J represent fork-join constructs that capture concurrent activities. Aircraft 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. A description of the network stations is provided in Table 1. Some stations are visited by all arriving aircraft, whereas others are visited according to specified probabilities.

To maintain model tractability, we impose the following simplifying assumptions:

1) The first assumption is the independence of customer behavior. The aircraft processing requirements are a function of the flight schedule, the aircraft manifest or en-route events, and are not linked explicitly to any flight line activities. Because the aircraft behave independently, we assume that the probability that an aircraft visits a particular service station (or a subnetwork of stations) is independent of the number and type of aircraft at the stations in the network.

2) Steady-state conditions are the second assumption. During wartime or a contingency, mobility operations are normally conducted around the clock over a period of days, weeks, or even months. Therefore, it is assumed that equilibrium conditions are maintained after some initial warm-up period.

3) The third assumption is single customer class. Normally, a mobility airfield processes different types of aircraft, which may have different service time distributions at certain network stations

Received 17 January 2001; revision received 26 April 2001; accepted for publication 6 May 2001.

*Lt Col, USAF, Department of Operational Sciences; currently Commander, AETC Studies and Analysis Squadron, 151 J Street East, Suite 2, Randolph AFB, Texas 78150; craig.willits@randolph.af.mil.

†Operations Research Consultant, Worldwide Emerging Technologies, 4001 Discovery Drive, Suite 130; dxdiet2@qwest.com; formerly Associate Professor of Operations Research, Department of Operational Sciences, Air Force Institute of Technology.

and different routing probabilities within the network. In the AAM these differences will not be explicitly modeled. Although this simplification may appear limiting, useful insights can still be gained by aggregating multiple aircraft classes through the appropriate selection of routing probabilities, service time distributions, and the arrival law. This approach will produce aggregate performance measures that can be used to gain fundamental insights into airfield capabilities.

III. Model Solution Through Product-Form Approximation

Exact calculation of network performance measures is not possible because of analytical complexity, but we can obtain approximate the performance measures accurately using a type of queueing network decomposition known as product-form approximation (PFA).[3] PFA methods use information about the flow in the network to construct a tractable network that has approximately the same steady-state behavior; the performance measures of the new network, which can be determined exactly using a number of available algorithms,[4,5] approximate those of the original network.

In a product-form approximation the original network is partitioned into a set of subnetworks, which are analyzed in isolation (that is, as independent networks) to get approximate throughput levels that are conditioned on subnetwork population. For each subnetwork an associated exponential server with load-dependent service rates is constructed; these service rates are set equal to the conditional throughput levels of the original subnetwork. The throughput levels are calculated in such a way that flow into and out of the exponential station closely approximates the flow behavior of the original subnetwork. A separable network is then formulated by replacing the subnetworks in the original network topology by the flow-equivalent servers; the performance measures of this network are used as approximations of those of the original network. The error in such an approximation originates from two sources: the assumption of exponential service and the approximation of the conditional throughputs through isolated analysis.

Building on the work of Dallery and Cao,[6] Baynat and Dallery identify four conditions that a network partition must satisfy in order for a PFA to be feasible (reasonably accurate)[3]:

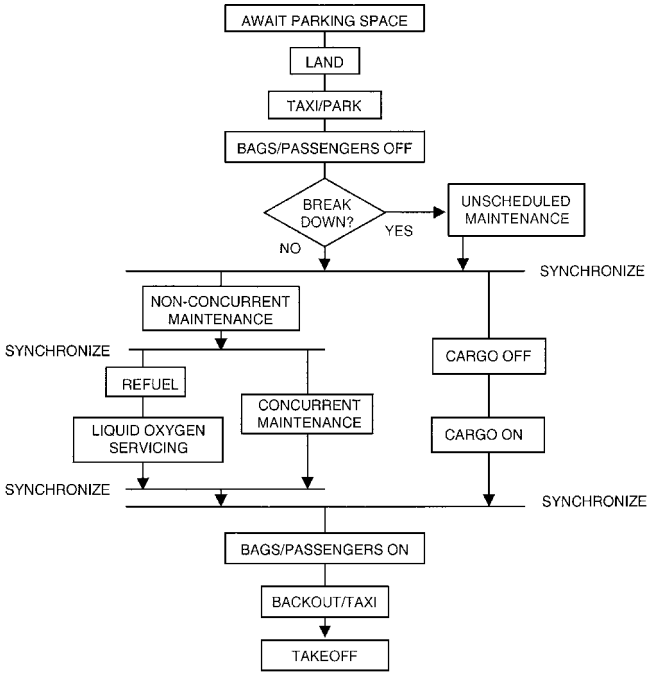

Fig. 1 Task precedence graph.

Table 1 AAM station descriptions

Station	Activity description	Visit probability
0	Interarrival time	1
1	Landing	1
2	Taxi/park	1
3	Maintenance (not concurrent with refueling)	≤ 1
4	Refuel	≤ 1
5	Liquid oxygen servicing	≤ 1
6	Maintenance (concurrent with refueling)	≤ 1
7	Cargo handling	≤ 1
8	Standard ground delay	1
9	Backout/taxi	1
10	Takeoff	1

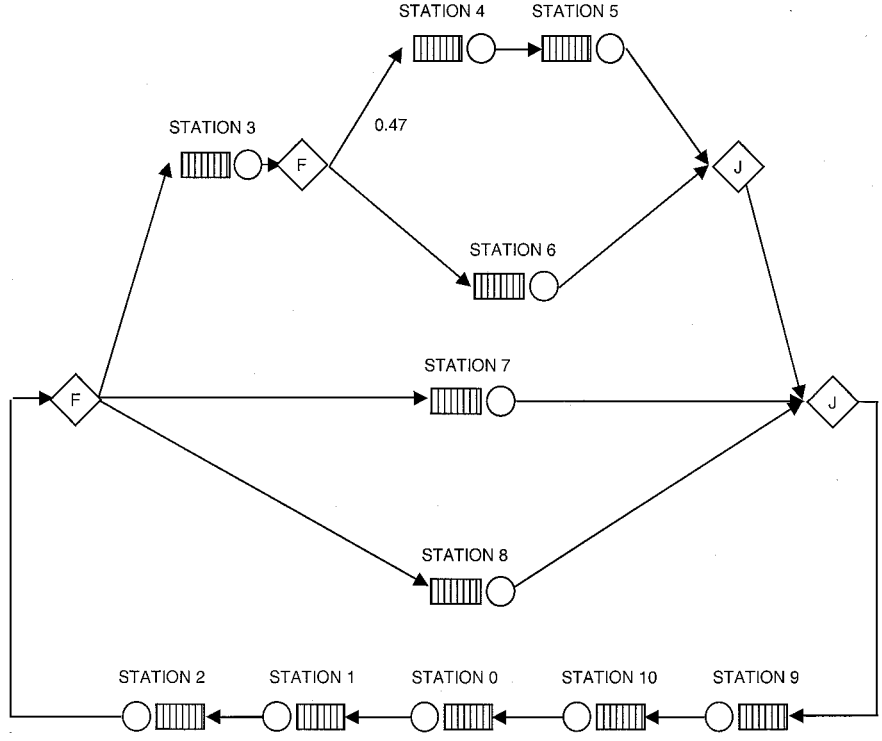

Fig. 2 AAM topology.

1) Customers enter and leave subnetworks as single entities.

2) The behavior of a subnetwork is independent of the behavior of its complement.

3) The routing between a subnetwork and its complement is independent of the number and distribution of customers the subnetwork.

4) Split customers (clones) do not transition between subnetworks.

If a fork-join queueing network (FJQN) has been feasibly partitioned using the preceding guidelines, then any subnetework containing a fork-join construct can be analyzed using either aggregation or Marie's method to get approximate conditional throughput levels. A discussion of these approaches follows, together with a summary of how each method can be used to analyze a FJQN.

A. Aggregation

Aggregation has its roots in the work of Avi-Itzhak and Heyman[7] and Chandy et al.[8,9] In this method the subnetwork to be isolated is analyzed as a closed, independent network. This network is formed by short-circuiting the subnetwork's complement (that is, removing the complement from the network). Approximate conditional throughput levels are obtained by calculating the throughput of this new subnetwork for fixed population levels. Chandy et al. showed that these conditional throughputs are exact if the original network is separable.[8] In the case where the network is nearly completely separable (that is, the behavior of the subnetworks is nearly mutually independent), the error induced by using aggregation will be small.[3]

If the conditional throughput levels of a FJQN are to be obtained using the aggregation technique, the isolated fork-join subnetwork (FJSN) would be formed by short-circuiting its complement, as in Fig. 3. Baynat and Dallery propose transforming this isolated network by creating separate customer chains for each clone and combining the fork node and join buffer into a multichain synchronization station with a deterministic zero service time and synchronized departures. The equivalent network is shown in Fig. 4.

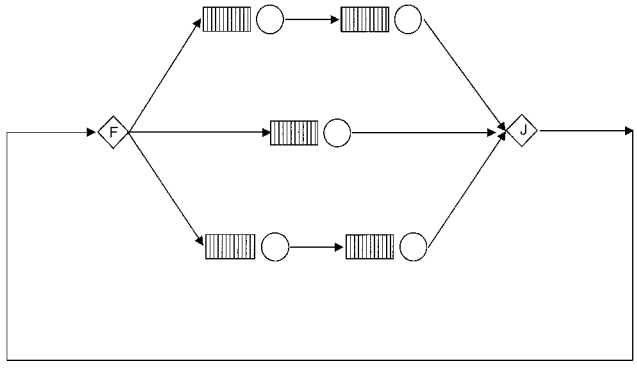

Fig. 3 Isolated fork-join subnetwork for the aggregation method.

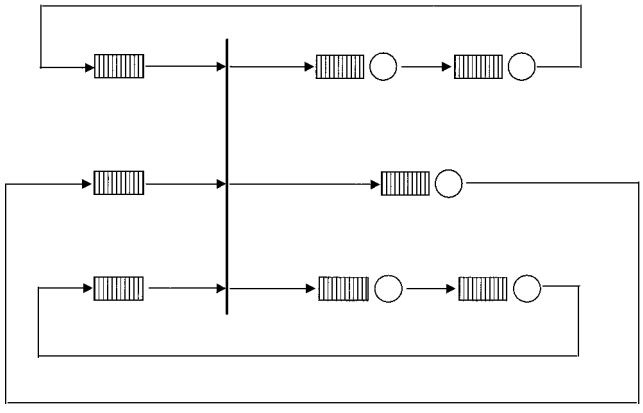

Fig. 4 Transformed subnetwork for the aggregation method.

B. Marie's Method

The central idea of Marie's method[10,11] is to analyze the subnetwork of interest as an isolated, open network with finite capacity and load-dependent Poisson arrivals. The load-dependent throughput levels of the isolated subnetwork become the load-dependent mean service rates of the associated exponential server.

Let N be the number of customers in the original closed network, and let n_i be the number of customers in subnetwork i ($0 \le n_i \le N$). Further, define $\lambda_i(n_i)$ and $\tilde{\nu}_i(n_i)$ as the arrival rate and the conditional throughput of customers at subnetwork i, $n_i = 0, \ldots, N$. Let $\mu_i(n_i)$ be the load-dependent service rate for the associated flow-equivalent exponential server, $n_i = 0, \ldots, N$.

Marie's method makes use of three sets of foundational equations. The first equation set, which is derived by applying the Marginal Local Balance Theorem,[12] establishes the throughput levels of the isolated subnetwork as a function of load n_i:

$$\tilde{\nu}_i(n_i) = \lambda_i(n_i - 1)\frac{\tilde{P}_i(n_i - 1)}{\tilde{P}_i(n_i)}, \qquad n_i = 1, \ldots, N \quad (1)$$

The probabilities $\tilde{P}_i(n_i)$ are the marginal probabilities that n_i customers occupy subnetwork i; these are found by analyzing the subnetwork in isolation as just described. We get the load-dependent service rates of the flow-equivalent exponential server by setting them equal to the throughput levels of the isolated subnetwork:

$$\mu_i(n_i) = \tilde{\nu}_i(n_i), \qquad n_i = 1, \ldots, N \quad (2)$$

The final equation set, which is also derived using the Marginal Local Balance Theorem, ensures local balance in the approximate product-form network:

$$\lambda_i(n_i) = \mu_i(n_i + 1)\frac{\hat{P}_i(n_i + 1)}{\hat{P}_i(n_i)}, \qquad n_i = 0, \ldots, N - 1 \quad (3)$$

The approximate occupancy probabilities $\hat{P}_i(n_i)$ are derived by analyzing the associated product-form network with any appropriate technique.[4,5]

Marie's algorithm solves Eqs. (1–3), for the conditional throughputs using fixed-point iteration. The algorithm is as follows:

1) Choose initial values $\mu_i(n_i)$ for $n_i = 1, \ldots, N$.

2) Calculate $\lambda_i(n_i)$ using Eq. (3).

3) Analyze the station in isolation to get $\tilde{P}_i(n_i)$, $n_i = 0, \ldots, N$.

4) Use Eq. (1) to get $\tilde{\nu}_i(n_i)$, $n_i = 1, \ldots, N$.

5) Calculate the load-dependent service rates $\mu_i(n_i)$ for the replacement server using Eq. (2).

6) Repeat steps 2 through 5 until the relative improvement in each $\mu_i(n_i)$ value is less than some specified tolerance.

The usual measure of improvement is the maximum relative change in the elements of the service rate vector:

$$\max_{i,n_i}\left|\frac{\mu_i^{(m)}(n_i) - \mu_i^{(m-1)}(n_i)}{\mu_i^{(m-1)}(n_i)}\right| < \varepsilon \quad (4)$$

where m is the iteration index and ε is the selected tolerance (typically set at 10^{-3} or 10^{-4}).

Marie's method is cited in several different studies as an accurate technique for decomposing nonseparable networks.[13–17] The method compares favorably to aggregation and provides superior estimates of expected queue lengths in many cases.[3] Bondi and Whitt find that Marie's method is the most accurate and stable of the decomposition techniques they examine.[16]

Baynat and Dallery have extended Marie's method so that it can be used to analyze closed networks with R (>1) chains. We employ multiple chains to distinguish between clone customers within a fork-join. The derivation is similar to that for the single-chain case, although each equation set must now be generated for each of the R chains. With the obvious extensions to the notation, the multiple-chain analogs to Eqs. (1–3) are

$$\tilde{\nu}_{ri}(n_{ri}) = \lambda_{ri}(n_{ri} - 1)\frac{\tilde{P}_{ri}(n_{ri} - 1)}{\tilde{P}_{ri}(n_{ri})}$$

$$n_{ri} = 1, \ldots, N_r, \qquad r = 1, \ldots, R \quad (5)$$

$$\mu_{ri}(n_{ri}) = \tilde{\nu}_{ri}(n_{ri}), \qquad n_{ri} = 1, \ldots, N_r, \quad r = 1, \ldots, R \quad (6)$$

$$\lambda_{ri}(n_{ri}) = \mu_{ri}(n_{ri}+1)\frac{\hat{P}_{ri}(n_{ri}+1)}{\hat{P}_{ri}(n_{ri})}$$

$$n_{ri} = 0, \ldots, N_r - 1, \qquad r = 1, \ldots, R \quad (7)$$

Marie's method for multiple chains is executed as follows[15]:

1) Choose initial values $\mu_{ri}(n_{ri})$ for $r = 1, \ldots, R$ and $n_{ri} = 1, \ldots, N_r$.

2) For $r = 1, \ldots, R$, calculate $\lambda_{ri}(n_{ri})$ using Eq. (7).

3) Analyze the station in isolation to get $\tilde{P}_{ri}(n_{ri})$, $r = 1, \ldots, R$ and $n_{ri} = 0, \ldots, N_r$.

4) Use Eq. (5) to get $\tilde{\nu}_{ri}(n_{ri})$, $r = 1, \ldots, R$ and $n_{ri} = 1, \ldots, N_r$.

5) Calculate the load-dependent service rates $\mu_{ri}(n_{ri})$ for the replacement server for chain r using Eq. (6).

6) Repeat steps 2 through 5 until the relative improvement in the $\mu_{ri}(n_{ri})$ values is less than some specified tolerance value.

The usual stopping test is similar to that given in Eq. (4), except that the maximization is also performed over the R chains in the network.

If Marie's method is to be used to decompose a FJQN, the isolated FJSN would be formed as an open, capacitated network with load-dependent Poisson arrivals; this would, in turn, be reformulated as the equivalent closed network shown in Fig. 5. Notice that the station representing the Poisson arrival process has mean service rate $\mu_0(n) = \lambda(N - n)$, $n = 1, \ldots, N$. Baynat and Dallery's transformation of this network is similar to the aggregation case, except that the join buffer is combined with the external Poisson arrival process to form a timed synchronization station. This station has mean service rate $\mu_0(n_0)$, where $n_0 = \min_r(n_{0r})$ and n_{0r} is the number of clones from chain r waiting in the join buffer. The equivalent network is shown in Fig. 6.

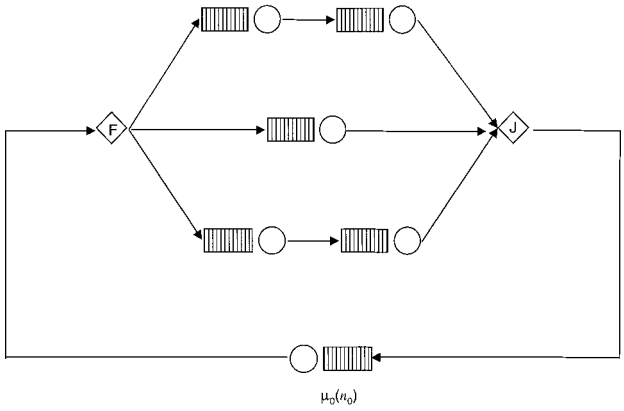

Fig. 5 Isolated fork-join subnetwork for Marie's method.

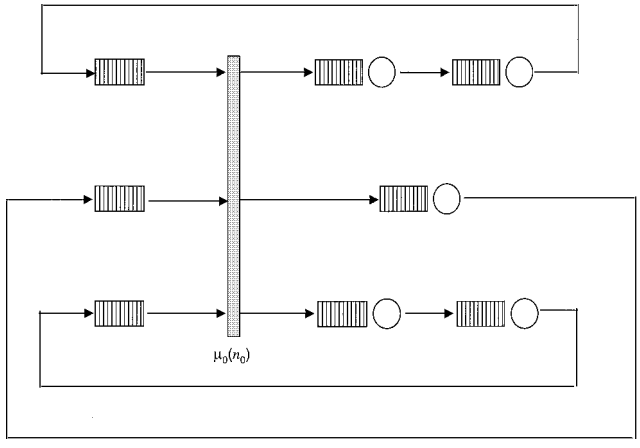

Fig. 6 Transformed subnetwork for Marie's method.

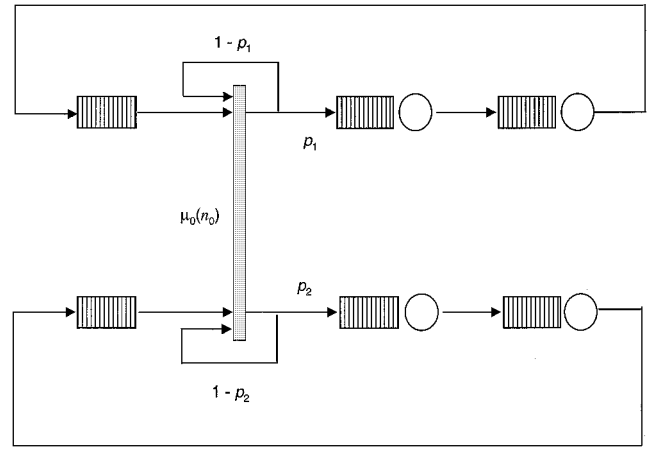

Fig. 7 Using the SC approximation with Marie's method.

IV. Short-Circuit Approximation

A. Description

Suppose we have a closed queueing network containing a FJSN with a probabilistic load pattern, meaning that a customer may completely bypass at least one embedded subnetwork, say the ith, with positive probability p_i. An intuitive way to model this behavior is to introduce feedback loops into the appropriate chains in the isolated, reformulated FJSN; these feedback loops allow a customer to bypass all stations in the chain and return immediately to the synchronization station. This strategy, which we will call the short-circuit (SC) approximation, is graphically illustrated in Fig. 7 (Marie's method has been used in the reformulation of the isolated FJSN).

The SC approximation requires an additional assumption that the customer clones in the isolated FJSN can match interchangeably. SC produces approximate results because the matching assumption may not be true for the original network model. An advantage of this approach is that the resulting expected increase in throughput induced by the assumption of interchangeability should partially offset the observed tendency of Marie's method to produce conservative approximations.[18]

B. Analyzing the Synchronization Station

The feedback loops just described can be dealt with in one of two ways: they can be incorporated into the internal Markov process of the synchronization station (internal feedback), or they can be left as part of the product-form approximation to the isolated FJSN (external feedback). In the latter case all that is required is to adjust the relative frequency of visits ("visit ratios") for the isolated FJSN. When the feedback loops are incorporated into the Markov process, however, the formulation is more complex. The following discussion develops the structure of the Markov process for the case of two clone chains within a fork-join construct. Extension to three or more chains is straightforward.

We first consider the case where aggregation is used to decompose the original network. Let n_i equal the number of chain i customers in the join buffer ($i = 1, 2$), and let (n_1, n_2) be the state of the synchronization station. Because the Markov process incorporates a join operation, the only feasible states are those for which $n_1 = 0$ or $n_2 = 0$ (or both). State transition behavior is complicated by the fact that one or both matching customers can return to the synchronization in zero time following a match.

Assume that $n_1 = 0$ and $0 \le n_2 \le N$ (where N is the network population). Define $\lambda_i(n_i)$ as the arrival rate of chain i customers, and let the bypass probability p_i be the probability that a chain i customer leaves the synchronization station. Further, let

$P_1(j) =$ Pr(chain 1 customer causes j chain 2 departures before leaving), $\qquad j < n_2$

and let

$P_1^0 =$ Pr(chain 1 customer causes n_2 chain 2 departures, and stays at station)

Table 2 State transitions for the aggregation method

Transitions to	Rate	Conditions
$(0, n_2 - j)$	$P_1(j)\lambda_1(0)$	$n_2 \in [0, N]$, $j \in [0, n_2]$
$(0, 1)$	$P_1^0\lambda_1(0)$	$n_2 \in [1, N]$
	$\lambda_1(0)$	$n_2 = 0$
$(0, n_2 + 1)$	$\lambda_2(0)$	$n_2 \in [0, N-1]$

The states that state $(0, n_2)$ transitions to, as well as the appropriate transition rates, are in Table 2.

We need to derive the probabilities $P_1(j)$ and P_1^0. To get $P_1(j)$, we condition on the number of feedback loops required to produce j chain 2 departures:

$$P_1(j) = \sum_{i=j}^{\infty} [\Pr(j \text{ chain 2 departures} \mid \text{chain 1 departure after } i\text{th loop}) \Pr(\text{chain 1 departure after } i\text{th loop})]$$

Clearly

$$\Pr(\text{chain 1 departure after } i\text{th loop}) = (1 - p_1)^{i-1} p_1$$

and

$$\Pr(j \text{ chain 2 departures} \mid \text{chain 1 departure after } i\text{th loop}) = \binom{i-1}{j-1} (1 - p_2)^{i-j} p_2^j$$

Therefore,

$$\begin{aligned} P_1(j) &= \sum_{i=j}^{\infty} \binom{i-1}{j-1} (1 - p_2)^{i-j} p_2^j (1 - p_1)^{i-1} p_1 \\ &= \sum_{i=1}^{\infty} \binom{i-1+j-1}{j-1} (1 - p_2)^{i-1} p_2^j (1 - p_1)^{i-1+j-1} p_1 \\ &= \frac{p_2^j (1 - p_1)^{j-2} p_1}{(1 - p_2)[1 - (1 - p_1)(1 - p_2)]^{j-1}} \sum_{i=1}^{\infty} \binom{i-1+j-1}{j-1} \\ &\quad \times [(1 - p_1)(1 - p_2)]^i [1 - (1 - p_1)(1 - p_2)]^{j-1} \\ &= \frac{p_2^j (1 - p_1)^{j-2} p_1}{(1 - p_2)[1 - (1 - p_1)(1 - p_2)]^{j-1}} \\ &\quad \times \left\{1 - [1 - (1 - p_1)(1 - p_2)]^{j-1}\right\} \end{aligned} \tag{8}$$

because each term in the infinite series is a negative binomial density.[19]

Because the probability that a chain 1 customer remains in the system after n_2 chain 2 departures is $(1 - p_1)^i$, we have that

$$P_1^0 = P_1(n_2)\frac{1 - p_1}{p_1} = \left(\frac{p_2^{n_2}(1 - p_1)^{n_2 - 1}}{(1 - p_2)[1 - (1 - p_1)(1 - p_2)]^{n_2 - 1}}\right) \times \left\{1 - [1 - (1 - p_1)(1 - p_2)]^{n_2 - 1}\right\} \tag{9}$$

The probabilities $P_2(j)$ and P_2^0 are easily derived by exchanging subscripts in Eqs. (8) and (9).

To efficiently derive the transition rate matrix Q for the Markov process, we first order the $2N + 1$ states as follows: $(N, 0), (N-1, 0), \ldots, (1,0), (0, 0), (0, 1), \ldots, (0, N-1), (0, N)$. Table 3 presents a complete description of the elements of the transition matrix.

When Marie's method is used, the process with feedback loops is somewhat simpler to formulate because of the nonzero delay after each synchronization. In this case states exist where both n_1 and n_2 are nonzero. The states that (n_1, n_2) transitions to, together with the appropriate transition rates, are shown in Table 4.

The transition rate matrix Q can be efficiently generated by ordering the states first on n_1, then on n_2: $(0, 0)$, $(0, 1), \ldots,$ $(N, N-1)$, (N, N). When this ordering scheme is followed, the nonzero entries in row s of Q can be generated according to the rules shown in Table 5.

Table 3 Column entries for row s of Q; aggregation method

Column index	Rate	Conditions
$s + j$	$P_2(j)\lambda_2(0)$	$s \in [1, N]$
$N + 2$	$P_2^0\lambda_2(0)$	$j = 1, \ldots, N + 1 - s$
$s - 1$	$\lambda_2(N + 1 - s)$	$s \in [2, N + 1]$
$s - j$	$P_1(j)\lambda_1(0)$	$s \in [N + 2, 2N + 1]$
N	$P_1^0\lambda_1(0)$	$j = 1, \ldots, s - N - 1$
$s + 1$	$\lambda_1(s - N - 1)$	$s \in [N + 1, 2N]$

Table 4 State transitions for Marie's method

Transitions to	Rate	Conditions
$(n_1 + 1, n_2)$	$\lambda_1(n_1)$	$n_1 < N$
$(n_1, n_2 + 1)$	$\lambda_2(n_2)$	$n_2 < N$
$(n_1 - 1, n_2)$	$p_1(1 - p_2)\mu_0(\min[n_1, n_2])$	
$(n_1, n_2 - 1)$	$(1 - p_1)p_2\mu_0(\min[n_1, n_2])$	$n_1, n_2 > 0$
$(n_1 - 1, n_2 - 1)$	$p_1 p_2 \mu_0(\min[n_1, n_2])$	

Table 5 Column entries for row s of Q; Marie's method

Column index	Rate	Conditions
$s + N + 1$	$\lambda_1(n_1)$	$n_1 < N$
$s + 1$	$\lambda_2(n_2)$	$n_2 < N$
$s - N - 1$	$p_1(1 - p_2)\mu_0(\min[n_1, n_2])$	
$s - 1$	$(1 - p_1)p_2\mu_0(\min[n_1, n_2])$	$n_1, n_2 > 0$
$s - N - 2$	$p_1 p_2 \mu_0(\min[n_1, n_2])$	

C. Extension to Nested FJQNs

Although nesting of fork-join constructs is not explicitly addressed in the open literature, it makes sense to deal with them by applying Baynat and Dallery's unified theory[3] in a hierarchical manner. All that is necessary is that the assumptions required by the unified theory be satisfied by the network partitions at all levels of the hierarchy.

For the purpose of illustration, suppose we have a FJQN with two FJSNs, one nested within the other. In this case hierarchical decomposition requires isolated analysis of structures at three levels, as illustrated in Fig. 8: 1) the nested fork-join construct (network 1); 2) the FJSN embedded in the nested construct, together with the outer synchronization station (network 2); and 3) the inner synchronization station (network 3). Other individual stations may need to be analyzed in isolation at any of these three levels.

D. Computational Experience

We conducted a numerical study to examine the performance of the SC approximation method when applied to FJQNs. Four candidate strategies were evaluated: 1) SC with internal feedback, using Marie's method to decompose the original network (SCMI); 2) SC with external feedback, using Marie's method to decompose the original network; 3) SC with internal feedback, using aggregation to decompose the original network; and 4) SC with external feedback, using aggregation to decompose the original network (SCAE). A variety of different network configurations was studied to ensure robust conclusions.[18] The results suggest that SC is a useful and generally highly accurate, approximation technique for closed FJQNs with probabilistic load patterns. This technique appears equally successful whether or not the network to be analyzed contains nested FJSNs.

In the nonnested case both SCAE and SCMI produce competitive approximations of expected throughput; this suggests that either approach would be useful if system-level performance measures are desired, particularly those measures that are relatively insensitive to

Table 6 Case study: AAM station parameters

Station	Activity description	Number of servers	Service discipline	Distribution type	Visit problem
0	Arrival	1	FCFS[a]	2-Coxian (3.79,0.65,0.41)[b]	1
1	Landing	1	FCFS	2-Erlang (0.033)	1
2	Taxi/park	8	Delay	2-Erlang (0.125)	1
3	Maintenance (not concurrent with refueling)	8	Delay	2-Erlang (0.083)	1
4	Refuel	6	FCFS	2-Erlang (0.983)	0.47
5	Liquid oxygen servicing	8	Delay	2-Erlang (0.45)	0.47
6	Maintenance (concurrent with refueling)	8	Delay	2-Erlang (0.5)	1
7	Cargo handling	3	FCFS	2-Erlang (0.946)	1
8	Standard ground delay[c]	8	Delay	2.34 (Deterministic)	1
9	Backout/taxi	8	Delay	2-Erlang (0.125)	1
10	Takeoff	1	FCFS	2-Erlang (0.033)	1

[a]FCFS ≡ first-come-first-served.
[b]In this paper a 2-stage Coxian distribution is denoted 2-Cox (μ_1, μ_2, α), where μ_i is the service rate in the ith stage and α is the transition probability between stages.
[c]Standard ground delay is built into the schedule by airlift planners to aid in system level planning; this number is typically a conservative deterministic estimate of the time required for ground processing.

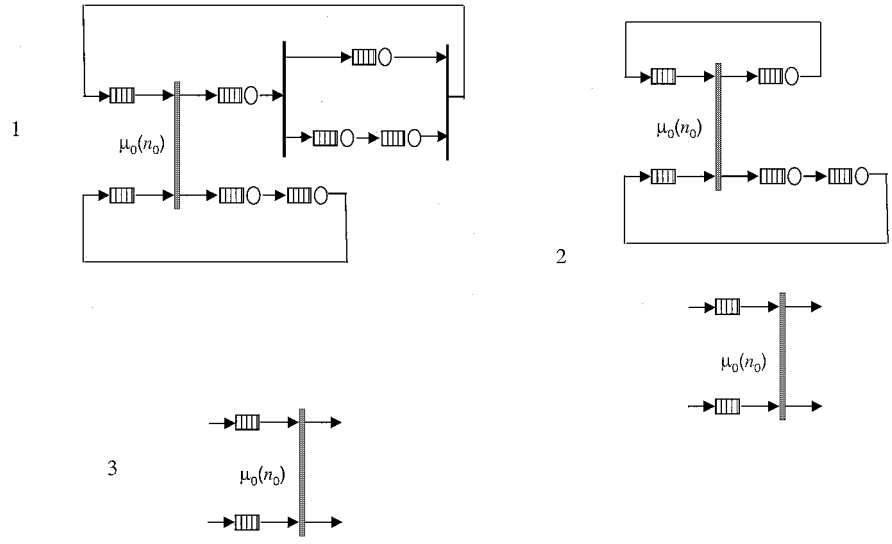

Fig. 8 Hierarchical decomposition of a nested FJQN.

higher moments of the service time distributions. SCMI is clearly the preferred method because it alone can provide accurate queue lengths for stations inside a fork-join structure. The SCMI method produced estimated throughputs and expected queue lengths with relative errors of less than two percent in most cases.

SCMI seems to suffer no degradation in performance when the network topology contains nested FJSNs. However, larger network populations may necessitate the use of a different stopping criterion for Marie's method. In some cases SCMI appears sensitive to service laws having coefficient of variation greater than one. However, such stations are generally not present in models of airfield operations.

V. Application and Results

We employed the AAM model in a case study of contingency operations at a representative mobility base. The airfield has a single runway and parking accomodations for up to eight aircraft. All resources other than cargo processing capability, aircraft refueling capability, and parking space are unconstrained. The parameters and aggregate service laws assumed for each service station are given in Table 6.

The aircraft arrival stream consisted of a list of aircraft by type, fuel load, cargo load, and so forth. The arrival stream data were preprocessed to determine the proportion of each type of aircraft, the proportion of aircraft needing fuel, and the mean and variance of the interarrival times. Service times for all ground processing tasks except refueling were determined from existing raw data; for cargo processing these were broken out by aircraft type. Pump rates, fuel truck travel times, and fuel line connect/disconnect times were provided so that refueling time could be determined by aircraft type; fuel hydrant and truck pump rates were aggregated by the proportion of each resource at the airfield.

The baseline AAM configuration was analyzed using the SCMI decomposition method. Three performance measures typically of interest to a mobility analyst were calculated:

1) The average airfield throughput (the average number of aircraft leaving the airfield each hour) = 1.1 departures per hour.

2) The average airfield response time (the average number of hours transpiring between aircraft arrival and departure) = 2.8 h.

3) The average number of aircraft on station = 3.0 aircraft.

Willits[18] presents more detailed results of the case study, including sensitivity analyses with respect to key input parameters.

Figure 9 illustrates the effect of variation in the average interarrival time on airfield throughput. Throughput increases moderately as the interarrival time decreases. Although one might be tempted to minimize the planned interarrival time (service time for the arrival station) to force a corresponding increase in throughput, we must recognize that this approach will cause the probability of airfield saturation to increase accordingly. Substantial command-and-control intervention would be required to divert planned arrivals away from the saturated airfield.

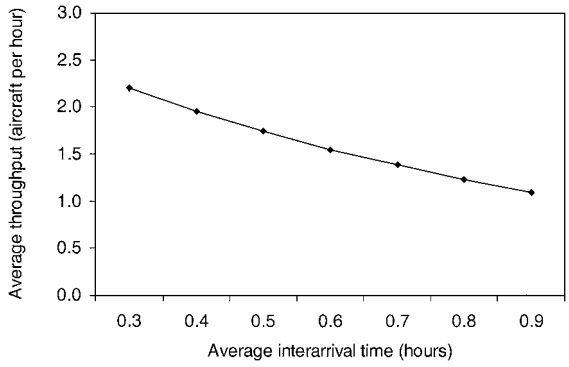

Fig. 9 Effect of mean interarrival time on airfield throughput.

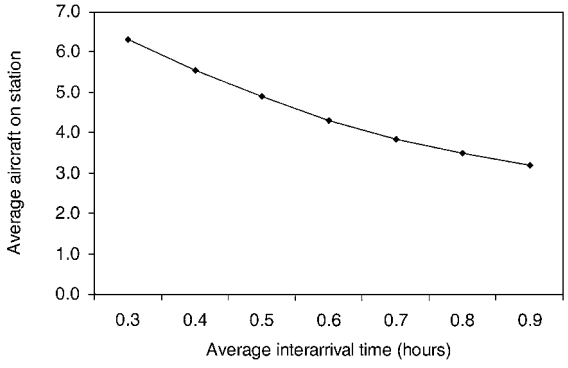

Fig. 10 Effect of mean interarrival time on aircraft on station.

Airfield response time was found to be insensitive to changes in the average interarrival time. This is probably caused by the domination of the response times by the standard ground time. In contrast, as the interarrival time decreases the average number of aircraft on station increases noticeably, but does not approach capacity (see Fig. 10). This implies that parking space is not a limiting factor at this airfield under these conditions.

We studied the effect of resource constraints by varying the maximum number of aircraft that could serviced for fuel or processed for cargo. All three performance measures were sensitive to a decrease in the number of cargo servers from the baseline value of three down to one, with throughput decreasing sharply and response time and aircraft on station showing a large increase. However, adjusting resources to permit more aircraft to simultaneously undergo cargo processing had no significant effect on any performance measure. Further, the performance measures were largely unaffected by adjustments to the maximum number of aircraft allowed to simultaneously refuel. Our sensitivity results imply that movable resources could be diverted from this airfield to address shortfalls elsewhere without substantially affecting airfield capability, as long as the airfield retains the capability to process at least two loads of cargo at the same time.

To investigate the effect of the standard ground delay on airfield operations, all performance measure values calculated during the sensitivity analysis were reevaluated with ground delay set to zero. When this was done, two effects were observed. First, airfield throughput increased slightly for low interarrival times, but the increase dropped to an insignificant level as the interarrival time increased. Second, both the airfield response time and the average number of aircraft on station showed a sharp decrease that was consistent in magnitude over all values of the mean interarrival time studied. This latter effect is intuitive, given the extent to which the ground delay dominates the other mean service times in the airfield flow (see Table 6). However, the insensitivity of airfield throughput to the change indicates that the savings in time are not great enough to forego the benefit of including the ground delay in system-wide planning (ground delay is desirable because it facilitates orderly management of the complete airfield system).

VI. Conclusions

The AAM provides a valuable supplement to high-resolution simulation modeling for gaining insights into mobility airfield capability. The speed and accuracy with which the model can be analyzed make it particularly useful for developing fundamental insights, performing sensitivity analyses, as possibly representing base level activity in a larger model of a complete airlift system. The model could also provide useful insights into other types of systems having multiserver queues, concurrent service activities, and general service time distributions (e.g., manufacturing).

To examine the accuracy of the analytical performance measures with respect to simulation, we calculated the relative error between each AAM performance measure and the analogous point estimate obtained by simulating the same network. Each simulation point estimate was refined until its associated 95% confidence interval had a half-width less than or equal to 10^{-2}. This half-width was used regardless of the magnitude of the point estimate, with the rationale that changes to queueing network parameters of lesser magnitude typically have little practical significance. All 318 performance measure data points calculated using the AAM had relative errors of less than 12%; 80% of these were 5% or less.

The software for the numerical investigation was implemented on a Digital Equipment Corporation (DEC) Alpha AXP 2100 Model 500MP workstation containing three 190 MHz DEC 21064 processors. To achieve the desired tolerance in the simulation point estimates, the typical simulation run required 35–45 min of elapsed system time. In contrast, virtually all of the analytical software runs produced near-immediate output.

Acknowledgment

The authors are grateful to the anonymous reviewers, whose thoughtful suggestions have significantly improved the article.

References

[1]Dietz, D., "Mean Value Analysis of Military Airlift Operations at an Individual Airfield," *Journal of Aircraft*, Vol. 36, No. 5, 1999, pp. 750–755.

[2]Gelenbe, E., and Pujolle, G., *Introduction to Queueing Networks*, 2nd ed., Wiley, New York, 1998, pp. 46–75.

[3]Baynat, B., and Dallery, Y., "A Unified View of Product-Form Approximation Techniques for General Closed Queueing Networks," *Performance Evaluation*, Vol. 18, No. 3, 1993, pp. 205–224.

[4]Bruell, S. C., and Balbo, G., *Computational Algorithms for Closed Queueing Networks*, North-Holland, New York, 1980, pp. 29–154.

[5]Conway, A. E., and Georganas, N. D., *Queueing Networks—Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*, Massachusetts Inst. of Technology Press, Cambridge, MA, 1989, pp. 20–212.

[6]Dallery, Y., and Cao, X., "Operational Analysis of Stochastic Closed Queueing Networks," *Performance Evaluation*, Vol. 14, No. 1, 1992, pp. 43–61.

[7]Avi-Itzhak, B., and Heyman, D., "Approximate Queueing Models for Multiprogramming Computer Systems," *Operations Research*, Vol. 21, No. 6, 1973, pp. 1212–1230.

[8]Chandy, K. M., Herzog, U., and Woo, I., "Approximate Analysis of General Queuing Networks," *IBM Journal of Research and Development*, Vol. 19, No. 1, 1975, pp. 43–49.

[9]Chandy, K. M., Herzog, U., and Woo, I., "Parametric Analysis of Queuing Networks," *IBM Journal of Research and Development*, Vol. 19, No. 1, 1975, pp. 36–42.

[10]Marie, R. A., "An Approximate Analytical Method for General Queueing Networks," *IEEE Transactions on Software Engineering*, Vol. 5, No. 5, 1979, pp. 530–538.

[11]Marie, R. A., Snyder, P. M., and Stewart, W. J., "Extensions and Computational Aspects of an Iterative Method," *Performance Evaluation Review*, Vol. 11, No. 4, 1982, pp. 186–194.

[12]Kant, K., *Introduction to Computer System Performance Evaluation*, McGraw–Hill, New York, 1992, p. 98.

[13]Baynat, B., and Dallery, Y., "A Decomposition Approximation for Closed Queueing Networks with Fork/Join Aubnetworks," *IFIP Transactions A (Computer Science and Technology)*, Vol. A-39, edited by M. Cosnard and R. Puigjaner, Elsevier, Amsterdam, 1993, pp. 199–210.

[14]Baynat, B., and Dallery, Y., "Approximate Analysis of Multi-Class Synchronized Closed Queueing Networks," *MASCOTS '95: Proceedings of the Third International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, edited by P. Dowd and E. Gelenbe, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 23–27.

[15]Baynat, B., Dallery, Y., and Ross, K., "A Decomposition Approximation Method for Multiclass BCMP Queueing Networks with Multiple Server Stations," *Annals of Operational Research*, Vol. 48, No. 1, 1994, pp. 273–294.

[16]Bondi, A. B., and Whitt, W., "The Influence of Service Time Variability in a Closed Network of Queues," *Performance Evaluation*, Vol. 6, No. 3, 1986, pp. 219–234.

[17]Dallery, Y., "Approximate Analysis of General Open Queueing Networks with Restricted Capacity," *Performance Evaluation*, Vol. 11, No. 3, 1990, pp. 209–222.

[18]Willits, C. J., "Nested Fork-Join Queuing Networks and Their Application to Mobility Airfield Operations Analysis," Ph.D. Dissertation, Department of Operational Sciences, Air Force Inst. of Technology, March 1997.

[19]DcGroot, M. H., *Probability and Statistics*, 2nd ed. Addison-Wesley, Reading, MA, 1986, pp. 258–262.