DECOMPOSITION ALGORITHMS FOR ANALYZING TRANSIENT PHENOMENA IN MULTICLASS QUEUEING NETWORKS IN AIR TRANSPORTATION

MICHAEL D. PETERSON

McKinsey & Company, Inc., Bedminster, New Jersey

DIMITRIS J. BERTSIMAS and AMEDEO R. ODONI

Massachusetts Institute of Technology, Cambridge, Massachusetts

(Received January 1993; revision received April 1994; accepted November 1994)

A previous paper (1992) by the same authors studied the phenomenon of transient congestion in landings at an airport and developed a recursive approach for computing moments of queue lengths and waiting times. This paper extends our approach to a network, developing two approximations based on the prior method. Both approaches work by using delay information estimated at one location to update arrival schedules at other points in the network. We present computational results for a simple 2-node network, comparing the performance of the approximations with an alternative simulation approach. The methods give similar results in light to moderate traffic but show a growing disparity under heavier traffic, where the algorithms underestimate the true magnitude of delay propagation relative to simulation. Finally, to illustrate the usefulness of the modeling, we show how the results may be used to explore the issue of interaction between airports. Although this particular application motivated development of the model, the method is, in principle, applicable to other multiclass queueing networks where service capacity at a station may be modeled as a Markov or semi-Markov process. The model represents a new approach for analyzing transient congestion phenomena in such networks.

Airport congestion and delay grew significantly during the 1980s, to the point where in 1990, 21 airports in the U.S. exceeded 20,000 hours of delay (55 aircraft hours per airport-day), with 12 more projected to exceed this total by 1997 (National Transportation Research Board 1991). Growing concern with these delays revived interest in operations research models that deal with air traffic congestion (for a review, see, e.g., Odoni, 1991).

From a queueing point of view, airport congestion problems are both challenging and important. Odoni and Roth (1983) and Roth (1981) showed that for a typical airport, the queue *relaxation time*—the time necessary to reach steady-state conditions—is significantly longer than the time over which the arrival process may be reasonably taken as constant. The work of Green, Kolesar and Svoronos (1991) and Green and Kolesar (1991, 1993) has also indicated the shortcomings of steady-state analysis under these conditions.

During the 1980s, air traffic congestion underwent a significant new development with the evolution of the hub-and-spoke network design now favored by almost all major carriers. This operational design offers significant economic gains to carriers, and compresses arrivals and departures at airports where connections occur ("hubs") into short intervals of time ("banks" of arrivals and departures), straining capacity, and creating a situation where traffic congestion and delays at the hub can have serious repercussions for delays throughout the network. Motivated by this important operational concern, Peterson, Bertsimas, and Odoni (1992) undertook a detailed study of congestion at a hub airport, employing a discrete-time model, which focused on changes in weather-related capacity. The study provided the beginnings of a new approach to this type of transient queueing problem, but ignored the potentially important phenomenon of interaction between airports in the network.

Modeling transient phenomena in the network context constitutes a still more difficult problem. Related research has been limited mainly to diffusion approximations (Kobayashi 1974, following the earlier work of Iglehart and Whitt 1970) and special case networks (Keilson and Servi 1990 on networks of $M/G/\infty$ queues). A different kind of approach was developed by Whitt (1983) with the queueing network analyzer. The approach presented in this paper is similar in spirit (though far different in content) to this latter work. We seek to extend the earlier work for one queue in isolation to the case of the network by way of approximating changes in arrival streams induced by earlier delays.

The model is motivated by the desire to study network effects of air traffic congestion, and in this context it provides important qualitative insight. This is illustrated

Subject classifications: Probability, stochastic model applications: semi-Markov and Markov models of airport capacity. Queues, transient results: approximations to compute performance measures in networks. Transportation, network models: study of congestion's effects in airline hub-and-spoke networks.

Area of review: STOCHASTIC MODELS AND THEIR APPLICATIONS.

0030-364X/95/4306-0995 $01.25

through two examples at the end of Section 4. The first examines the "coupling" between a pair of congested major airports. In this context, the model is envisioned as a strategic-level planning tool to assist schedulers in assessing the effects of hub "connectivity"—the percentage of aircraft sharing multiple hubs—on the propagation of delay from one hub to another. A second example illustrates how the model is useful in deciding how much "slack" time should be added to the period an aircraft spends between successive flight legs. The purpose of these slack times, which have been used increasingly by airlines during the past 10 years, is to reduce the impact that delays early in a day may have on flights later on. Weather variability and the attendant variability of airport capacities from day to day make it necessary to consider carefully the tradeoff between, on one hand, the efficient utilization of aircraft and crews and, on the other, the reliable execution of advertised schedules. One major airline (American) is currently in the process of addressing this same problem by developing a simulation of its flight schedule, using historical records of the probability distribution of flight times, including delays, between individual pairs of airports (FitzGerald 1993). By contrast, the approach we describe here relies on a numerical queueing model that computes the delays at each airport from the demand profile and a probabilistic model of airport capacities. The model is a planning tool in the sense that it is concerned with developing a strategy, *vis-à-vis* time allowed between flights, that takes into consideration the full range of demand/capacity relationships that may prevail during a season at each major airport in a network. In executing the resulting schedule on a daily basis, an airline will make tactical adjustments through such actions as cancellation of some delayed flights and substitution of late-arriving aircraft by spare aircraft.

Beyond the air transportation context, our work comprises another step in developing a body of knowledge in the transient analysis of multiclass queueing networks. Although air transportation problems motivated this work from the start, the techniques presented are applicable for general queueing networks with time-varying arrivals which may be approximated as deterministic over short periods and with time-varying and serially correlated service times. Our approximation methods parallel, in a crude sense, those of the QNA due to Whitt. We hope that the work will stimulate further thinking in one of the more difficult subfields of operations research.

The remainder of the paper is organized as follows. In Section 1 we review briefly the model previously developed for queueing at a single station and describe the network context of the present problem. In Sections 2 and 3 we outline two decomposition approaches which exploit the single-queue model. Section 2 describes a relatively simple method in which downstream arrivals are adjusted according to expected upstream waiting times. Section 3 describes a more involved approach which uses second moment information about delays to give a stochastic description of downstream arrival rates. Section 4 employs these approximation methods together with a simulation procedure on a 2-hub network. We provide computational results for several test problems; these illustrate nicely the behavior of the network and show where the approximations do and do not work well. To conclude this section, we briefly illustrate the usefulness of the model in studying network effects in air transportation. Section 5 summarizes our main conclusions and suggests areas for further work.

1. THE BASIC MODEL

The unit of analysis in this queueing problem is the airport, where incoming arrivals require service at three stations: a landing runway, a gate, and a departure runway. The landing operation, in particular, is subject to wide variations in capacity due to weather conditions. For purposes of this paper, we focus on landings as the source of delays and consider the system of one or more landing runways as constituting a single server.

Consider a network of airports $n = 1, \ldots, N$. For airport n, the aircraft arrival process is highly time varying, especially in the case of a hub, where traffic is concentrated into "banks," intervals of highly concentrated arrivals or departures. We assume *initially* that this process is deterministic but time varying. We divide time into short intervals of fixed length Δt and let the number of aircraft demanding to land at airport n in period k be given by the parameter λ_k^n. Within period k these arrivals are assumed to constitute a uniform flow. Landing capacity during a given interval k is assumed to be in one of $S(n)$ states $i = 1, \ldots, S(n)$ corresponding to service rates $\mu_1^n, \mu_2^n, \ldots, \mu_{S(n)}^n$, where

$$\mu_1^n < \mu_2^n < \cdots < \mu_{S(n)}^n.$$

These states correspond to the capacities available under different configurations and weather conditions. In an application of this model to Dallas-Fort Worth (DFW) (Peterson, Bertsimas and Odoni), we found $S(n) = 6$ to be an adequate number of capacity states.

For a given capacity state i at airport n we assume a random duration T_i^n which follows an arbitrary discrete distribution

$$P_i^n(m) = \Pr\{T_i^n = m\},$$

the probability of a capacity μ_i period lasting for precisely m intervals of length Δt. Upon exiting a state i, the capacity process enters another state $j \neq i$ with probability p_{ij}. In the computational results section, we will employ a more specialized version of the model in which holding times are geometrically distributed and capacity follows a Markov chain. This assumption is *not* necessary for model development, but the Markov model

does give substantially better *computational* performance. Results reported for our study at DFW indicate that model outputs are largely insensitive to the distributional assumption.

Our assumptions imply that during any interval k, a single queue in isolation behaves like a deterministic flow process. That is, if q_k is the length of the queue at the end of some period k, then the queue length one period later is the maximum of 0 and the values $q_k + \lambda_{k+1} - \mu_i$ for $i \in \{1, \ldots, S\}$. Define the state of the airport at any time to be $\{i, m\}$, where i identifies current capacity and m is the time (in intervals) for which that capacity has prevailed. The combined age-capacity process is Markov with transition probabilities

$$\begin{aligned}\tilde{p}_{ij}(m) &\triangleq \Pr((i, m) \to (j, 1)) \\ &= \Pr[T_i = m | T_i \geq m] p_{ij}, \quad j \neq i, \\ \tilde{p}_{ii}(m) &\triangleq \Pr((i, m) \to (i, m+1)) \\ &= \Pr[T_i \geq m+1 | T_i \geq m]. \end{aligned} \tag{1}$$

We next define the random variables:

$Q_k \triangleq$ the queue length at end of interval k;
$W_k \triangleq$ the waiting time at end of interval k;
$C_k \triangleq$ the capacity state at end of interval k;
$A_k \triangleq$ the age of current capacity state at end of interval k;
$T_i \triangleq$ the random lifetime of capacity state i.

For mean queue length we introduce the notation

$$\begin{aligned}&\mathcal{Q}_k(l, i, m, q) \triangleq E[Q_k | Q_l = q, C_l = i, A_l = m] \\ &k = 1, \ldots, K, \quad i = 1, \ldots, S, \quad m = 1, \ldots, M, \\ &l \leq k, \quad q = 1, \ldots, q_{max}(k, i), \end{aligned} \tag{2}$$

where $q_{max}(k, i)$ is the maximum attainable queue length at the end of period k, given that at that time the capacity state is i. This obeys the recursion

$$q_{max}(k, i) = [q_{max}(k-1) + \lambda_k - \mu_i]^+ \tag{3}$$

where $q_{max}(k) \triangleq \max_i q_{max}(k, i)$ and $x^+ = \max(x, 0)$. Similarly, for waiting times we employ the notation

$$\mathcal{W}_k(l, i, m, q) \triangleq E[W_k | Q_l = q, C_l = i, A_l = m]. \tag{4}$$

We write the second moment analogs of (2) and (4) as $\mathcal{Q}_k^2(l, i, m, q)$ and $\mathcal{W}_k^2(l, i, m, q)$, respectively.

Let $(x \wedge y)$ denote $\min(x, y)$. The quantities $\mathcal{Q}_k(l, i, m, q)$, $\mathcal{Q}_k^2(l, i, m, q)$, $\mathcal{W}_k(l, i, m, q)$, and $\mathcal{W}_k^2(l, i, m, q)$ can be calculated recursively, (Peterson, Bertsimas and Odoni). We repeat here the basic equations:

$$\begin{aligned}&\mathcal{Q}_k(l, i, m, q) \\ &= \sum_{j \neq i} \tilde{p}_{ij}(m) \mathcal{Q}_k(l+1, j, 1, (q + \lambda_{l+1} - \mu_j)^+) \\ &\quad + \tilde{p}_{ii}(m) \mathcal{Q}_k(l+1, i, m+1, (q + \lambda_{l+1} - \mu_i)^+), \end{aligned} \tag{5}$$

$$\begin{aligned}&\mathcal{Q}_k^2(l, i, m, q) = \sum_{j \neq i} \tilde{p}_{ij}(m) \mathcal{Q}_k^2 \\ &\qquad \cdot (l+1, j, 1, (q + \lambda_{l+1} - \mu_j)^+) \\ &\qquad + \tilde{p}_{ii}(m) \mathcal{Q}_k^2(l+1, i, m+1, \\ &\qquad (q + \lambda_{l+1} - \mu_i)^+), \end{aligned} \tag{6}$$

$$\begin{aligned}&\mathcal{W}_k(l, i, m, q) \\ &= \sum_{j \neq i} \tilde{p}_{ij}(m) [\mathcal{W}_k(l+1, j, 1, (q + \lambda_{l+1} - \mu_j)^+)] \\ &\quad + \tilde{p}_{ii}(m) \mathcal{W}_k(l+1, i, m+1, (q + \lambda_{l+1} - \mu_i)^+), \end{aligned} \tag{7}$$

$$\begin{aligned}&\mathcal{W}_k^2(l, i, m, q) \\ &= \sum_{j \neq i} \tilde{p}_{ij}(m) [\mathcal{W}_k^2(l+1, j, 1, (q + \lambda_{l+1} - \mu_j)^+)] \\ &\quad + \tilde{p}_{ii}(m) [\mathcal{W}_k^2(l+1, i, m+1, (q + \lambda_{l+1} - \mu_i)^+)], \end{aligned} \tag{8}$$

$$\begin{aligned}&\mathcal{W}_k(k, i, m, q) \\ &= \sum_{j \neq i} \tilde{p}_{ij}(m) \left[\left(\frac{q}{\mu_j} \wedge 1 \right) + \mathcal{W}_k(k, j, 1, (q - \mu_j)^+) \right] \\ &\quad + \tilde{p}_{ii}(m) \left[\left(\frac{q}{\mu_i} \wedge 1 \right) + \mathcal{W}_k(k, i, m+1, (q - \mu_i)^+) \right], \end{aligned} \tag{9}$$

$$\begin{aligned}&\mathcal{W}_k^2(k, i, m, q) \\ &= \sum_{j \neq i} \tilde{p}_{ij}(m) \left[\left(\frac{q}{\mu_j} \wedge 1 \right)^2 + 2 \left(\frac{q}{\mu_j} \wedge 1 \right) \mathcal{W}_k(k, j, 1, \right. \\ &\qquad \left. (q - \mu_j)^+) + \mathcal{W}_k^2(k, j, 1, (q - \mu_j)^+) \right] \\ &\quad + \tilde{p}_{ii}(m) \left[\left(\frac{q}{\mu_i} \wedge 1 \right)^2 + 2 \left(\frac{q}{\mu_i} \wedge 1 \right) \mathcal{W}_k(k, i, m+1, \right. \\ &\qquad \left. (q - \mu_i)^+) + \mathcal{W}_k^2(k, i, m+1, (q - \mu_i)^+) \right], \end{aligned}$$

with boundary conditions

$$\mathcal{Q}_k(k, \cdot, \cdot, q) \equiv q, \tag{11}$$

$$\mathcal{Q}_k^2(k, \cdot, \cdot, q) \equiv q^2, \tag{12}$$

$$\mathcal{W}_k(k, \cdot, \cdot, 0) \equiv 0, \tag{13}$$

$$\mathcal{W}_k^2(k, \cdot, \cdot, 0) \equiv 0. \tag{14}$$

For a single airport in isolation, (5)–(14) allow us to compute recursively the expectations and variances for queue lengths and waiting times at the end of each interval, based on given initial conditions. This can be achieved with computational complexity $O(S^2K^2MQ_{max})$, where S is the number of capacity states, K the total number of time intervals, M an upper bound on the memory argument m, and $Q_{max} \triangleq \max_k q_{max}(k)$ is the highest attainable queue length over all periods. In the Markov case, the dimension m is unnecessary, and the running time reduces to $O(S^2K^2Q_{max})$.

Return now to the network of airports $n = 1, 2, \ldots, N$. On this network let there be a set $\mathcal{A}$ of aircraft numbered $v = 1, 2, \ldots, V$. Divide the operating day into

periods of length Δt, numbered as $k = 1, 2, \ldots, K$. Each aircraft v has an itinerary

$$\mathcal{I}(v) \triangleq \{(i_m^v, t_m^v, s_m^v)\} \quad m = 1, 2, \ldots,$$

where

$i_m^v \triangleq$ the mth stop on itinerary of aircraft v;
$t_m^v \triangleq$ the scheduled arrival time at mth stop for aircraft v;
$s_m^v \triangleq$ the slack time between stops $m-1$ and m for aircraft v.

Aircraft *slack* between stops $m-1$ and m is the amount of time available to the aircraft at stop $m-1$ beyond the minimal time necessary to turn it around. In the network, schedules are no longer exogenous and deterministic, as delays at one airport affect the schedules at others. In the terminology of queueing theory, the system is a multiclass queueing network, with the classes being the different aircraft with their individual itineraries. Service capacity at each airport is an autocorrelated stochastic process described by a semi-Markov process or Markov chain. Thus, our task is to describe the transient behavior of a multiclass queueing network with autocorrelated service rates at each node. This high degree of complexity suggests that approximation methods are necessary.

2. A SIMPLE DECOMPOSITION APPROACH

A first approximation approach is based on the following idea. Suppose that at the start of the day, one knows the schedules for all aircraft operating in the network. Under the assumption that delays are zero at the outset of the day, the schedule for the initial period of the day is fixed. Hence the first period demands are fixed, and mean queue lengths and waiting times for each airport during this period may be determined by applying (5)–(14) to each airport. The resulting expected waiting times for period 1 are estimates of the delay encountered by all aircraft scheduled to land in this period. Taking into account the slack which these aircraft have in their schedules and updating future arrival streams accordingly, one then fixes demand for the next period, calculates the resulting new expected waiting times, and so forth.

More formally, let d^v represent the current cumulative delay for aircraft v, i.e., as aircraft v proceeds through its itinerary, d^v is the current amount by which it is behind schedule. Further define the terms

$\mathcal{A}(n, k) \triangleq$ the set of aircraft scheduled to land at n in period k;
$E[W_k^n] \triangleq$ the mean waiting time for an aircraft arriving at airport i at end of period k;
$\lambda_k^i \triangleq$ the number of scheduled arrivals at airport i during period k.

The arrival times t_m^v are real numbers which represent times within the integer time periods. Time $t = 0$ is the start of the operating day. Let $\kappa(t)$ be the function which takes real-time values into their corresponding periods:

$$\kappa(t) = \left\lceil \frac{t}{\Delta t} \right\rceil.$$

The scheduled arrival rates $\{\lambda_k^n\}$ are determined from the sets of aircraft $\mathcal{A}(n, k)$ which are in turn determined by the itineraries $\mathcal{I}(v)$:

$$\lambda_k^n = |\mathcal{A}(n, k)| \tag{15}$$

$$\mathcal{A}(n, k) = \{v: (n, t, s) \in \mathcal{I}(v) \text{ for some } s \text{ and } \kappa(t) = k\} \tag{16}$$

Consider an aircraft which arrives at airport n at some time t during period k. An estimate of this aircraft's waiting time to land is the convex combination of expected waiting times at the end of periods $k-1$ and k,

$$\alpha E[W_{k-1}^n] + (1-\alpha)E[W_k^n], \tag{17}$$

with the weight α determined by whether t lies closer to the end of period k or $k-1$:

$$\alpha = \frac{\kappa(t) - t}{\Delta t}. \tag{18}$$

Not all of this delay is necessarily propagated to later points in the system, however, because of slack and cumulative delay, d^v should be adjusted to reflect this fact. To illustrate, let the above aircraft's next scheduled stop (stop $m+1$) be n' at time t', and suppose that from the current stop until the next stop there is an available slack of s'. Prior to the mth stop, the aircraft's cumulative delay was d^v; thus its new scheduled arrival time is given by

$$t' + (d^v + \alpha E[W_{k-1}^n] + (1-\alpha)E[W_k^n] - s')^+. \tag{19}$$

In words, the aircraft's delay into its next stop is the maximum of zero and current delay plus new delay less schedule slack. Algorithm 1, based on this simple idea, is thus summarized as follows:

1. Initialize arrival schedules $\mathcal{A}(n, k)$ and set all delays $d^v = 0$.
2. For all periods $k = 1, \ldots, K$ and airports $n = 1, \ldots, N$
   - Set $\lambda_k^n = |\mathcal{A}(n, k)|$
   - Calculate $E[W_k^1], \ldots, E[W_k^N]$ from (5)–(14).
   - For each aircraft arriving in period k, update expected delay via (19).
   - Update arrivals $\mathcal{A}(n, k)$ according to the updated delays.

The full algorithm is given in the Appendix.

In computing expected waiting times, we must aggregate aircraft and compute the level of demand at each airport, while in the schedule-updating procedure we disaggregate to the level of individual aircraft. To make this procedure efficient, we employ the data structure of Figure 1. The arrival sets $\mathcal{A}(i, k)$ are singly-linked lists of aircraft indexed by current destination i and arrival period k. The number of aircraft records hung from a

particular location in the data structure constitutes the demand rate for that period. Once delays are updated, each affected aircraft record is rehung from a new part of the arrival matrix. With this data structure the inner updating loop requires only $O(V)$ time, so the bottleneck consists of repeated calls to a subroutine for computing expected waiting times. Because for each time period k the algorithm must recalculate all of the preceding expected waiting times, overall complexity is $O(KNU)$, where U is the complexity of the single hub algorithm. In Peterson, Bertsimas and Odoni it was shown that for a Markov model of capacity (i.e., where capacity durations are geometric), $U = O(S^2K^2Q_{max})$. Thus, if the Markov capacity model is specified with S capacity states, overall complexity for Algorithm 1 is $O(NS^2K^3Q_{max})$. For general duration times, running times are multiplied by a factor M which constitutes a practical bound on the range over which the hazard rate for the duration is nonconstant ($M = 20$ for the Dallas hub).

The presence of the additional factor K arises from the fact that the recursion is restarted from time 0 at each new period. This duplication of effort could be avoided if it were possible to store within the single hub algorithm the end conditions of iteration k as initial conditions for iteration $k + 1$. However, this would mean storing the joint probabilities for queue length and capacity, and computing these probabilities requires $O(Q_{max})$ times as much effort as for the expectation alone (see Peterson, Bertsimas and Odoni). A more practical improvement is to have the recursion restart only every m periods, where m is the minimum number of periods any aircraft has between scheduled stops. While in the original implementation, the number of iterations performed within the recursive algorithm is $K(K + 1)/2$, under this new scheme it is

$$m + 2m + 3m + \cdots + Gm + K' = G(G + 1)m/2 + K',$$

where $G = \lfloor K/m \rfloor$ and

$$K' = \begin{cases} K & \text{if } Gm < K \\ 0 & \text{otherwise.} \end{cases}$$

This modification alone leads to substantial savings. The number of iterations is reduced by a factor

$$\frac{K(K + 1)/2}{m\lfloor K/m \rfloor(\lfloor K/m \rfloor + 1)} \geq \frac{K(K + 1)}{K(K/m + 1)} = \frac{K + 1}{K/m + 1}.$$

In the case $K = 80$, for example, a value of $m = 10$ implies that the number of iterations is reduced from 3,240 to 360, one-ninth of the former number. Because of the high computational requirements of the network problem, the speed advantage of the Markov model over the semi-Markov model (nongeometric service durations) is meaningful, while case study results suggest that the results are insensitive to the input distribution. For these reasons, we will focus on the Markov formulation of capacity for the remainder of the paper.

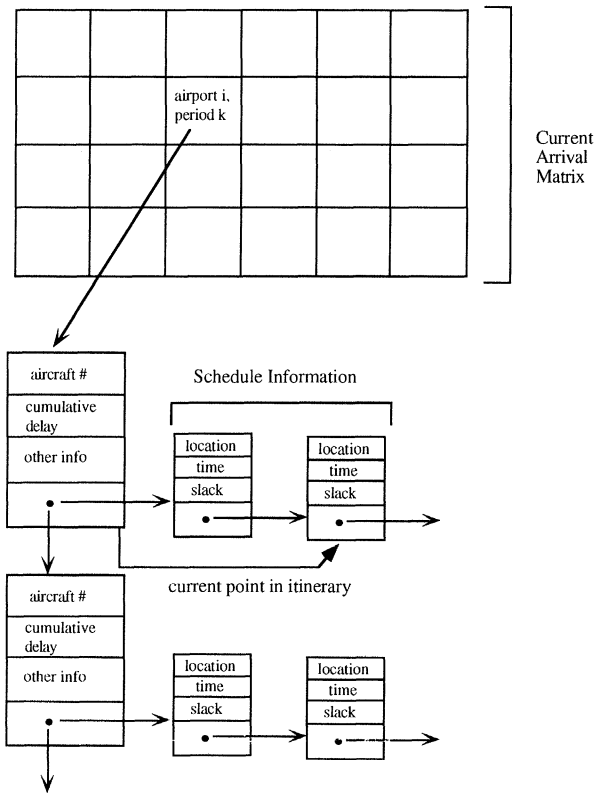

Figure 1. Data structure used in network congestion algorithms.

3. AN ALGORITHM WITH PROBABILISTIC UPDATING

The updating scheme of the previous section takes deterministic arrival streams and uses expected waiting time information to convert them into new deterministic arrival streams. A better method might take into account the *variance* in the waiting times, as well as the mean, in specifying information about future arrivals rates. These arrival rates are thus specified probabilistically rather than deterministically.

For airport n at period k, let the expectation and variance of the waiting time be μ and σ^2, respectively. Let $f_k^n(w)$ be a density for the waiting time W_k^n estimated (see below) from these parameters. Given this density and the schedule slacks, we may characterize (probabilistically) the next arrival period of each aircraft $v \in \mathcal{A}(n, k)$. Specifically, we compute numbers $p_v(0), \ldots, p_v(C)$ and $k_v(0), \ldots, k_v(C)$ such that the next period in which aircraft v will land is $k_v(i)$ with probability $p_v(i)$. Here, the parameter C is a practical upper bound on the number of periods of delay possible. Figure 2 illustrates this phenomenon of traffic "splitting."

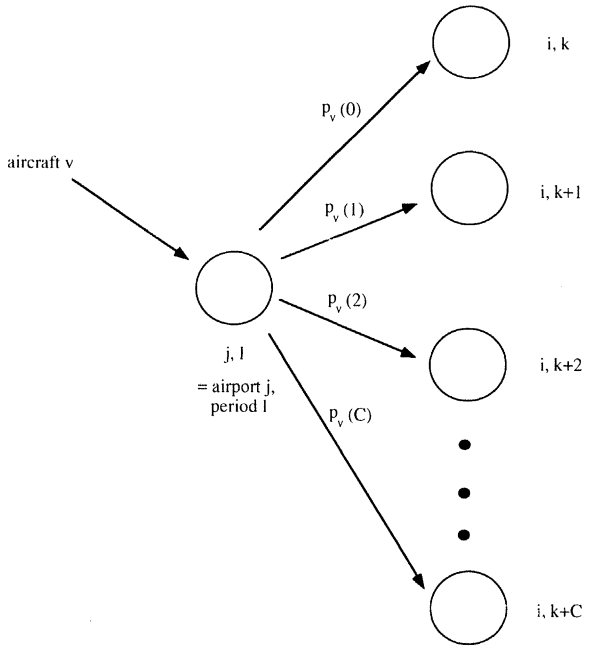

Figure 2. The traffic splitting phenomenon: alternative future aircraft paths depend upon delay encountered. The numbers $\{p_v\}$ indicate probabilities.

Next, define the stochastic arrival quantities

$\Lambda(n, k) \triangleq$ = the number of arrivals at airport n in period k.

For some user-specified number R (representing the number of possible values taken by the random variables) suppose that we may estimate numbers $\gamma_k^n(1), \ldots, \gamma_k^n(R)$ and $\lambda_k^n(1), \ldots, \lambda_k^n(R)$ such that

$$\begin{aligned} &\Pr\{\Lambda(n, k) = \lambda_k^n(1)\} = \gamma_k^n(1), \\ &\Pr\{\Lambda(n, k) = \lambda_k^n(2)\} = \gamma_k^n(2), \\ &\qquad\vdots \\ &\Pr\{\Lambda(n, k) = \lambda_k^n(R)\} = \gamma_k^n(R). \end{aligned} \tag{20}$$

where $\sum_i \gamma_k^n(i) = 1$. This variability in the arrival rates is easily incorporated into the recursion for expected queue lengths and waiting times; the innermost loop of the recursion is rewritten to take the expectation over all possible values of Λ_k. For example (c.f. Kobayashi),

$$\begin{aligned} &{}^{\circ}W_k(l, i, m, q) \\ &= \sum_{r=1}^{R} \gamma_{l+1}^r \Big[\tilde{p}_{ii}(m)\,{}^{\circ}W_k(l+1, i, m+1, \\ &\qquad \cdot (q + \lambda_{l+1}^r - \mu_i)^+) + \sum_{j \neq i} \tilde{p}_{ij}(m)\,{}^{\circ}W_k \\ &\qquad \cdot (l+1, j, 1, (q + \lambda_{l+1}^r - \mu_j)^+) \Big]. \end{aligned} \tag{21}$$

This recursion produces future waiting time estimates, leading to new densities, new arrival probabilities, and so on. Thus, the previous description suggests an alternative algorithm, Algorithm 2, that will be described in detail.

Figure 2 suggests another important point. Because of uncertainty in delays, an aircraft landing at a particular place and time takes one of many future paths. Ideally, we would like to keep track of all such future paths and thus be able to assign probabilities to all realizations of the sets $\mathcal{A}(n, k)$. Unfortunately, the computational complexity inherent in this task is overwhelming because of the large number of such paths—$O(C^{\zeta(v)})$ for each aircraft v, where $\zeta(v)$ is the number of airports in v's itinerary. Thus, while we can reflect the splitting phenomenon in assigning probabilities to the different values $\lambda_k^n(\cdot)$, we must limit the realizations of the sets $\mathcal{A}(n, k)$. To accomplish this, we again update each aircraft's *cumulative delay* by a convex combination of $E[W_k]$ and $E[W_{k-1}]$. Unlike Algorithm 1, Algorithm 2 allows a partial modeling of the splitting phenomenon (through the λ_k^n's), but only *part* of the phenomenon is captured, i.e., the immediate effect of delay uncertainty on the next period's arrival rates. This splitting is *not* reflected when aircraft schedules are updated.

As outlined, the second decomposition algorithm requires four separate procedures:

1. Estimation of the densities $f_k^n(w;\ \mu(i, k),\ \sigma^2(i, k))$ for the waiting times at each station and period, given the estimates of mean and variance computed in the recursion.
2. Translation of these density functions into probabilistic descriptions of future arrival periods for each aircraft, as given in the parameters $p_v(0), \ldots, p_v(C)$ and $k_v(0), \ldots, k_v(C)$.
3. Translation of the individual aircraft parameters $p_v(0), \ldots, p_v(C)$ and $k_v(0), \ldots, k_v(C)$ into simple discrete distributions for the random variables $\Lambda(n, k)$.
4. Updating of aircraft itineraries and airport arrival lists.

The fourth of these procedures was described in Section 2. The first three are described in further detail in what follows, and a summary of the algorithm is given in the Appendix.

3.1. Obtaining Waiting Time Densities

Without prior assumptions, estimation of the densities $f(w)$ on the basis of knowing only two moments is problematic. In the case of a single airport, a simple simulation of the capacity process (from the Markov chain) suggests a starting point. Under deterministic arrival assumptions, the simulation of period capacities yields the matrix of observations

$$\mathbf{W} = \{W_k^m\},$$

where W_k^m is the waiting time at the end of period k for the mth (independent) simulation. A sample histogram

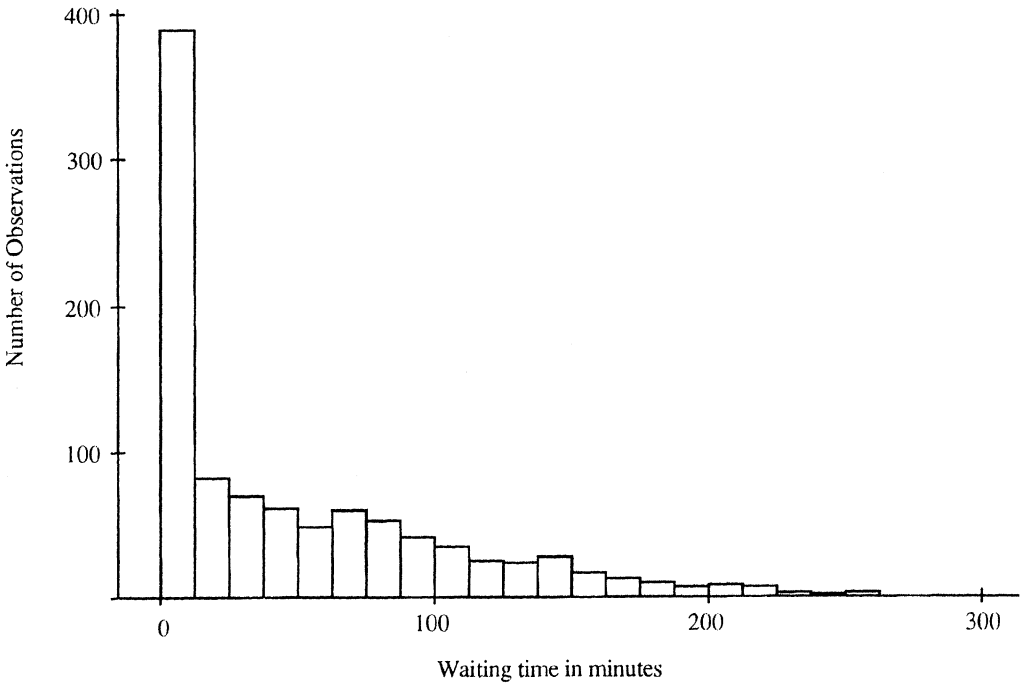

Figure 3. Histogram from simulated waiting times in a single queue.

for the waiting time for period 50 (for the case of a constant arrival rate $\lambda = 60$ per hour and $\rho \approx 0.85$) is illustrated in Figure 3. Note the presence of a substantial probability mass at the minimum value (in this case, 0). Values above this minimum follow an approximately exponential distribution, and probability plots (not shown) confirm this. The results suggest an approximate mixed distribution for the waiting times W_k^n:

$$\Pr\{W_k^n = w_{min}(n, k)\} = \delta$$

$$\Pr\{W_k^n \leq w | w > w_{min}(n, k)\} = 1 - e^{-\nu(w - w_{min}(n,k))}. \quad (22)$$

The parameters $w_{min}(n, k)$, usually but not always 0, can be calculated directly from the recursion in a manner similar to that for the parameters $q_{max}(n, k)$. The parameters δ and ν must be determined by solving the pair of equations (omitting subscripts)

$$\delta w_{min} + (1 - \delta) \int_{w_{min}}^{\infty} w \nu e^{-\nu(w - w_{min})}\, dw = E[W],$$

$$\delta (w_{min})^2 + (1 - \delta) \int_{w_{min}}^{\infty} w^2 \nu e^{-\nu(w - w_{min})}\, dw = E[W^2]. \quad (23)$$

In terms of the mean $\bar{w}$ and variance σ^2 we obtain the solution (omitting subscripts)

$$\delta = \frac{\sigma^2 - (\bar{w} - w_{min})^2}{\sigma^2 + (\bar{w} - w_{min})^2}, \quad (24)$$

$$\nu = \frac{2(\bar{w} - w_{min})}{\sigma^2 + (\bar{w} - w_{min})^2}. \quad (25)$$

Note that δ is always less than 1 and will be nonnegative provided that

$$\frac{\sigma^2}{(\bar{w} - w_{min})^2} \geq 1.$$

In the typical case where w_{min} is zero, this is equivalent to the condition that the squared coefficient of variation for waiting times exceeds 1. Only in rare instances of the tests presented shortly was this condition found not to hold. In those cases, the parameter δ was set to 0 and the entire distribution was assumed to be exponential.

3.2. From Densities to Schedules

Given estimated densities for W_k^n for all points n in the network, the next step in the procedure is to infer probabilities for the immediate future paths of all aircraft $v \in \mathcal{A}(n, k)$. For any such aircraft, let (n', t', s') be the scheduled next stop (stop $m + 1$) on its itinerary. The earliest period in which this aircraft's next landing may actually take place is

$$k_v(0) = \kappa(t' + [d^v + w_{min} - s']^+).$$

This is the earliest period at which this aircraft could next land, reflecting the minimum waiting time achievable at this stop (usually 0). Accordingly, the greatest amount of delay this aircraft can endure at i and have this next arrival period remain unaltered is

$$\begin{aligned} w(0) &= \max\{w' : \kappa(t' + [d^v + w' - s']^+) = k_v(0)\} \\ &= \{w' : t' + d^v + w' - s' = k_v(0)\Delta t\} \\ &= k_v(0)\Delta t - t' - d^v + s', \end{aligned}$$

where d^v is its cumulative delay prior to the mth stop. The probability that the aircraft's next scheduled period is $k_v(0)$ is

$$p_v(0) = \int_{w_{min}}^{w(0)} f(w;\ \mu,\ \sigma^2)\ dw. \tag{26}$$

If $w_{min} = 0$, which is usually the case, $k_v(0)$ corresponds to the outcome that zero additional periods of delay are added to aircraft v at this stop. When the waiting time density is approximated by (22) with $w_{min} = 0$, (26) becomes

$$p_v(0) = \delta + (1 - \delta)[1 - \exp(-\lambda w(0))].$$

Letting $w(1) = w(0) + \Delta t$, the probability of the next scheduled period being $k_v(1) \equiv k_v(0) + 1$ is

$$p_v(1) = \int_{w(0)}^{w(1)} f(w;\ \mu,\ \sigma^2)\ dw, \tag{27}$$

and, in general, the probability of c additional periods of delay is

$$p_v(c) = \int_{w(c-1)}^{w(c)} f(w;\ \mu,\ \sigma^2)\ dw, \tag{28}$$

where $w(c) = w(0) + c\Delta t$. These expressions take the appropriate specific forms when the distribution (22) is substituted.

For practical reasons, it is necessary to choose some upper bound C on the number of periods of delay to allow. Hence

$$p_v(C) = \int_{w(C-1)}^{\infty} f(w;\ \mu,\ \sigma^2)\ dw.$$

Together with the numbers $\{k_v(c)\}$, the probabilities $\{p_v(c)\}$ then constitute a probabilistic description of the next period in which aircraft v will demand to land.

3.3. Characterizing Arrivals

To translate the numbers $\{p_v(c)\}$ into a probabilistic description of the future demand rates $\Lambda(n, k)$, define the random variable

$$X_{n'l,nk}(v) \triangleq \begin{cases} 1 & \text{if } v \in \mathcal{A}(n', l) \text{ is delayed such that its} \\ & \text{next stop will be } n \text{ at period } k \\ 0 & \text{otherwise.} \end{cases}$$

This random variable denotes the "contribution" of an arrival at one place and time to the arrival rate at a future place and time. Note that if the next stop of $v \in \mathcal{A}(n', l)$ is n, then

$$\Pr\{X_{n'l,nk}(v) = 1\} = p_v(k - l).$$

In words, for aircraft $v \in \mathcal{A}(n', l)$, the probability that it will contribute to the landing demand at airport n during period k (assuming that n is its next scheduled stop) is $p_v(k - l)$.

The random variables $X_{n'l,nk}(v)$ provide the necessary connection between aircraft and arrival rates via

$$\Lambda(n, k) = \sum_{n'=1}^{N} \sum_{l<k} \sum_{v=1}^{V} X_{n'l,nk}(v). \tag{29}$$

In words, this says that the arrival rate at (n, k) is the sum of all contributions from previous points in the itineraries (e.g., see Figure 4). Thus, the random variables $\{\Lambda\}$ are sums of Bernoulli random variables. Defining

$$NL(v, k) \triangleq = \text{next destination of aircraft } v \text{ after period } k$$

the expectation is easily obtained as

$$E[\Lambda(n, k)] = \sum_{n'=1}^{N} \sum_{l<k} \sum_{v=1}^{V} E[X_{n'l,nk}(v)].$$

$$= \sum_{n'=1}^{N} \sum_{l<k} \sum_{v:NL(v,l)=n} p_v(k - l). \tag{30}$$

Obtaining the variance of $\Lambda(n, k)$ is not straightforward because the terms of the sum are not independent. Aircraft delayed at earlier points in the day may share the same source for those delays, so that their contributions to future demands may be correlated. On the other hand, diversity in scheduling and slack weaken this dependence. For the sake of tractability, we make the approximation that the contributions are approximately independent and write

$$Var[\Lambda(n, k)]$$

$$\approx \sum_{n'=1}^{N} \sum_{l<k} \sum_{v:NL(v,l)=n} p_v(k - l)(1 - p_v(k - l)). \tag{31}$$

This approximation agrees quite closely with simulation results.

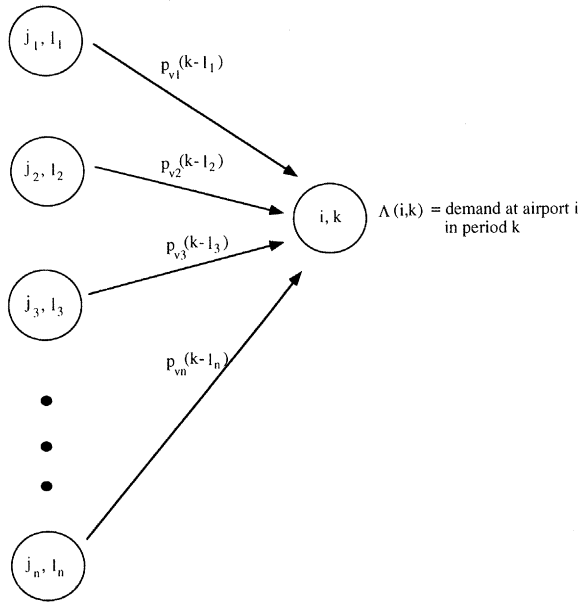

Figure 4. Updating downstream arrivals in Algorithm 2: early arrivals and delays contribute to demands later in the day.

The specification of approximate distributions for the $\{\Lambda(n, k)\}$ is the final step in translating aircraft delays into arrival rate information. Again, we confront the issue of estimating a distribution from only two moments. The form (29) suggests a normal form based on the central limit theorem idea, though convergence may not be good due to nonindependence of the terms of the sum. Simulation results indicate that for early periods of the day where there are fewer terms in the sum, unusual skewness patterns are possible (see Figure 5). These patterns disappear later in the day. While this phenomenon is cause for some concern, test runs also indicate a considerable degree of insensitivity to the demand rate distribution. We retain the normality assumption while acknowledging its imperfections.

Although Algorithm 2 involves considerably more modeling work than Algorithm 1, its computational complexity is only slightly higher, $O(RKNU)$, where R is the user-specified number of values used in the approximate distribution for the arrival rates, and U is the complexity of the single hub recursive algorithm with deterministic input. If the Markov capacity model is specified with S capacity states, the overall complexity is $O(RNS^2K^3Q_{max})$.

4. TESTING THE DECOMPOSITION MODELS

Both Algorithms 1 and 2 are suitable for a general network (i.e., not necessarily hub and spoke). However, without the streamlining suggested at the end of Section 2, running times are somewhat high for large networks. For a simple 2-airport network with $K = 60$ periods at each station, Algorithm 1 takes about 10 minutes on an Ultrix DECsystem 5900 workstation, while Algorithm 2 takes about 30 minutes. With the reduction in calls to the recursion achieved by the streamlining procedure, there is roughly tenfold improvement in these figures. Even with this improvement, modeling a full-size network of a large airline (400+ nodes) is a somewhat daunting problem.

The problem is well suited to parallel computation, with different processors handling the individual nodes and a central processor controlling the bookkeeping of aggregation and disaggregation. However, further simplification is clearly desirable. In this respect, note that from the perspective of a single air carrier serving a hub-and-spoke network, delays at the *hubs* have far greater implications for disruption of schedule than delays at the *spokes*. This observation suggests that we reduce the network to include only the hubs, tracking only those aircraft belonging to the hub carrier and incorporating spoke information in setting itineraries. We treat other arrivals as fixed and assume that congestion delays other than those emanating from the hubs are *negligible*. All internal flights in the collapsed network (see Figure 6) appear to take place between hubs, but flight times vary to reflect intermediate spoke stops. The reduced model collapses a large airline's network from 400+ nodes to perhaps 5 or 6 but still captures essential behavior.

4.1. Testing Procedure

The network of Figure 6 will serve as a testing ground for the decomposition algorithms; its simple structure readily allows experimentation and interpretion of the results in terms of the demand and capacity behavior. Table I summarizes eight test cases, which differ with

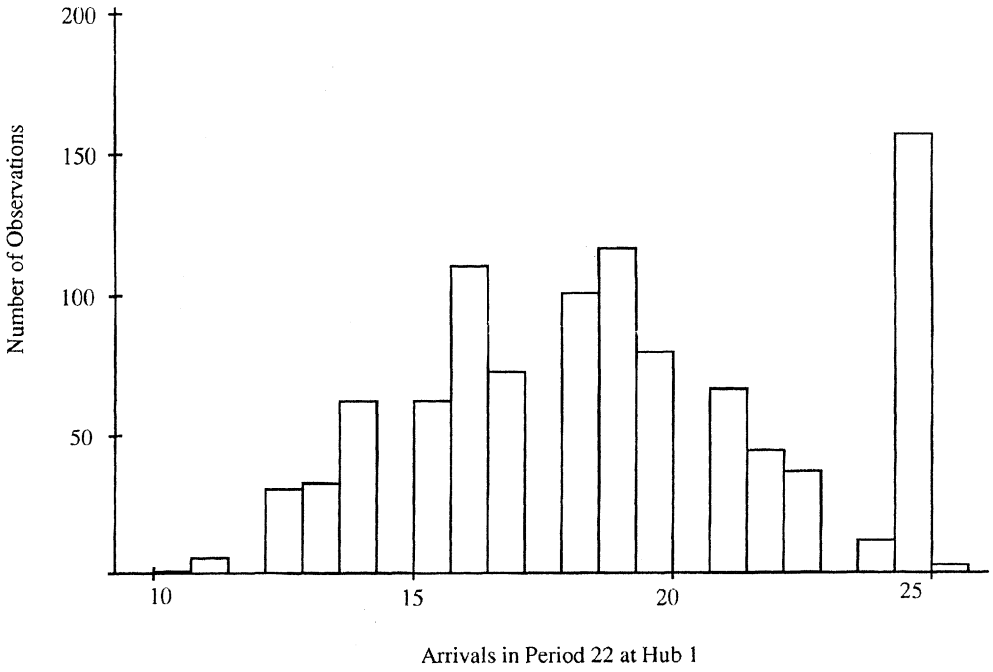

Figure 5. Histogram of $\Lambda(1, 22)$ obtained from simulation. Unusual skewness patterns such as this one may occur in the early part of the day when the contributing prior arrivals are still largely deterministic.

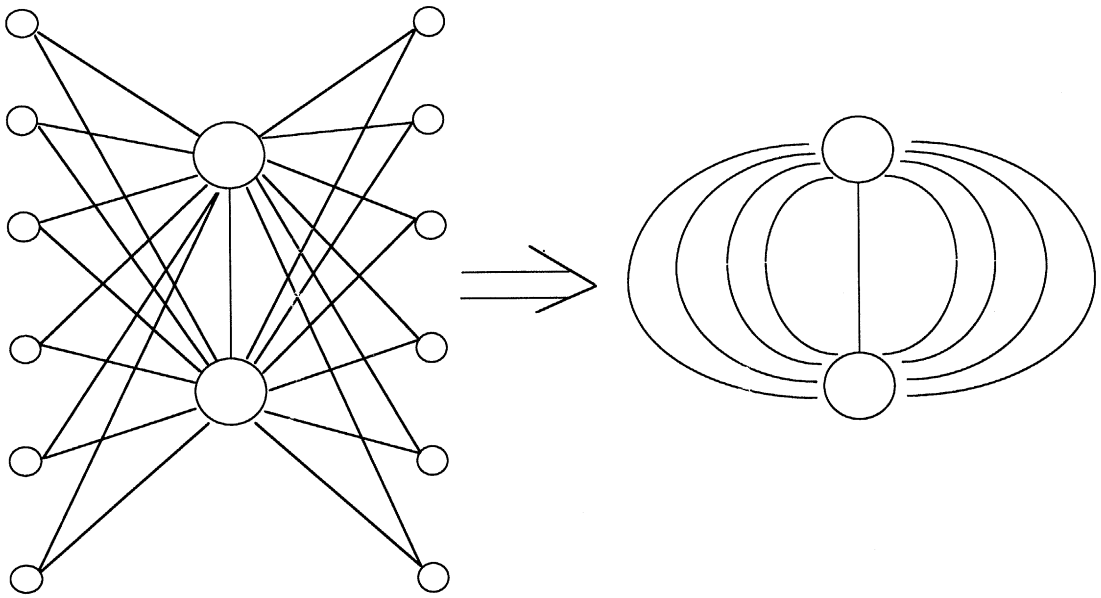

Figure 6. Two-hub test network obtained from larger hub-and-spoke network.

respect to the number of banks (i.e., the peak arrival periods), the degree of separation between banks, the percentage p of flights which visit different hubs (rather than the same hub) on alternate visits, the time-averaged traffic intensity ρ, the amount of schedule slack, and the initial capacity conditions. The parameter p is a measure of how each node is tied to the performance at the other. A value of $p = 1$ implies a fully connected network (all flights alternate between the hubs).

Demand and capacity data for case 1 closely resemble those at Dallas-Fort Worth. The three capacity states are associated with poor, medium, and good weather conditions. The steady-state probabilities associated with these three states are 0.07, 0.10, and 0.83, which corresponds to the self-transition probabilities (p_{ii}) of 0.92, 0.9, and 0.98. The demand data are simulated as follows. First, each "internal aircraft" (i.e., belonging to the host carrier) is randomly assigned to one of the two hubs, and a first arrival time is chosen from one of the first 5(?) arrival banks. Subsequent locations and scheduled landing times for the aircraft are then chosen according to the value p such that the resulting demand profile closely resembles that for DFW in March 1989 (these data were employed by the authors in the earlier study of DFW). Aircraft slacks (the cushion available prior to each aircraft trip) take values in the range of 15–20 minutes between stops at hubs, depending on the distance to the intervening spoke.

In case 2, schedules for internal aircraft are simulated in a similar fashion, but the result but the demand pattern groups aircraft into banks of 30-minutes duration at each hub, with relatively short periods of 15 minutes in between. Peak demands are higher than in case 1, while capacities are slightly lower (the underlying Markov chain has steady-state probabilities 0.26, 0.21, 0.53, and p_{ii} values 0.9, 0.8, and 0.95). This second experiment represents the extreme of tight scheduling, with slack reduced to 5 minutes per stop. Case 3 reports results for

Table I

Test Run Information (note that traffic intensities ρ are based on that part of the schedule which does not include the runout period at the end of the day; steady state indicates that initial capacities occur according to the steady-state probabilities of the Markov chain)

Case	Number of Banks	Bank Space	p	ρ	Slack	Initial Capacities
1	(DFW)	30 mins (avg)	0	0.5	15–20 mins.	Low/high
2	12	15 mins.	0.5	0.9	5 mins.	Steady state
3	—	—	0.5	0.8	5 mins.	Steady state
4a	10	30 mins.	0	0.7	5 mins.	Low/high
4b	10	30 mins.	1	0.7	5 mins.	Low/high
5a	10	30 mins.	0.5	0.7	5 mins.	Steady state
5b	10	30 mins.	0.5	0.7	10 mins.	Steady state
5c	10	30 mins.	0.5	0.7	15 mins.	Steady state
5d	10	30 mins.	0.5	0.7	20 mins.	Steady state

a continuous demand pattern at the two airports (no pronounced peaks), with capacity the same as in case 2. Cases 4 (a and b) and 5 (a–d) are concerned with the effects of slack and connectivity on schedule reliability. Both have a traffic pattern like that of case 2, but with lower landing demand and greater bank separation.

The first three cases compare the results of the decomposition algorithms with the results from simulation. This simulation procedure is based on the same Markov chain model of capacity but does not employ the approximation procedures for delay propagation implied by (17)–(19) and (20)–(31); instead, it performs updates according to realized delays. In the terminology of Law and Kelton (1991), this is a terminating simulation, with each replication lasting one operating day of $k = 60$ periods (80 for case 1). Capacity switches between the three states according to the simulated progress of a Markov chain, with initial conditions sampled according to the steady-state probabilities. Queue length is determined each period as a deterministic flow (given demand and capacity), and the waiting time is calculated from the subsequent capacity path. For example, if the queue is 20 at a given time and the capacity remains at 80 per hour for the next half hour, the wait is calculated to be 15 minutes. Scheduled demand is updated according to cumulative delay as determined by simulated waiting times and slack (the principal departure from the approximations). Replication $j (j = 1, \ldots, N)$ of the simulation produces the waiting times $W_{j1}, W_{j2}, \ldots, W_{j,K}$. For each period k, the random variables $W_{jk}, j = 1, \ldots, N$ are i.i.d. (Law and Kelton), and an unbiased estimator of the waiting time is given by

$$\bar{W}_k = \sum_{j=1}^{N} W_{jk}, \tag{32}$$

with an estimated standard error

$$\hat{\sigma}_k = \sqrt{\frac{\sum_{j=1}^{N} [W_{jk} - \bar{W}_k]^2}{N - 1}}. \tag{33}$$

A simulation of $N = 10{,}000$ replications gives a relative error of approximately 1% for sample mean waiting times and ensures that rare sample paths are represented (for example, an instance of 60 consecutive periods of state 1 would occur approximately 17 times in 10,000 replications). Results are reported for an implementation in C run on an Ultrix DECsystem 5900 workstation.

4.2. Results and Discussion

Cases 1–3: Model Comparison. Figure 7 shows the expected waiting times at hub 1 for test cases 1–3, as predicted by the simulation and the two network approximations. Each period's estimated waiting time from simulation has a standard error of approximately 1%, which implies an approximate 95% confidence interval of ±2% relative to the reported value. Because periods clearly are not independent, one cannot make comparable confidence statements about the path taken as a whole (this is the "multiple comparisons problem—see Law and Kelton, p. 569); however, this level of precision is surely adequate for discussing qualitative behavior.

The top part of the figure shows case 1, which approximates the situation at the Dallas-Fort Worth hub. The earlier study (Peterson, Bertsimas and Odoni) considered this hub *in isolation* without taking into account delays encountered elsewhere in the network; the estimates obtained in that case are also included in the top graph. As may be readily seen, there is fairly close agreement between all four curves. This situation owes to the fact that expected waiting times are approximately equal to slack values (15–20 minutes), so that the amount of delay carried over is small. The simulation case deviates slightly from the others after hour 10 because propagation in that case only is based on individual realizations of waiting time rather than on the expected value. Simulation thus reflects a degree of "tail behavior" which the others do not. The resulting longer delay propagations shift more traffic to the later part of the day when demand is low, smoothing the overall demand profile and reducing expected queueing delays at those times. This smoothing influence of the network is small, however, compared with the influence of the peaked arrival pattern itself on waiting times; in other words, for this type of schedule, the network influence does not predominate.

The situation is different in cases 2 and 3 (the center and bottom of Figure 7). In case 2, the expected waiting times (30–40 minutes) are high relative to aircraft slack (5 minutes), and banks are closely spaced. As the day progresses, propagated delays become significant and are reflected in an increasing gap between the decomposition algorithms and the simulation. As in case 1, the two algorithms' "update by expected waiting time" strategy does not fully reflect the shift of traffic to the end of the day and the resulting schedule smoothing. This smoothing reduces expected waiting times by as much as 30% for some periods. The mean deviation of algorithmic output from simulation output may be estimated by the "standard error"

$$s = \sqrt{\frac{\sum_{k=1}^{K} (X_k - Y_k)^2}{K - 1}},$$

where X_k is the waiting time value predicted by the algorithm for period k and Y_k the corresponding value for the simulation. For Algorithms 1 and 2, these numbers are 6.11 and 5.69, respectively, reflecting a 0–30% discrepancy over the course of the day. Case 3 (demand continuous over the day—no peaks) also shows this magnitude of deviation (see the bottom of Figure 7).

A glance at the waiting time profiles for cases 1 and 2 (top and center of the figure) indicates that waiting time profiles are much smoother in the latter case. In fact, the peaked pattern of demand is barely visible in case 2a; propagated delays are numerous enough to

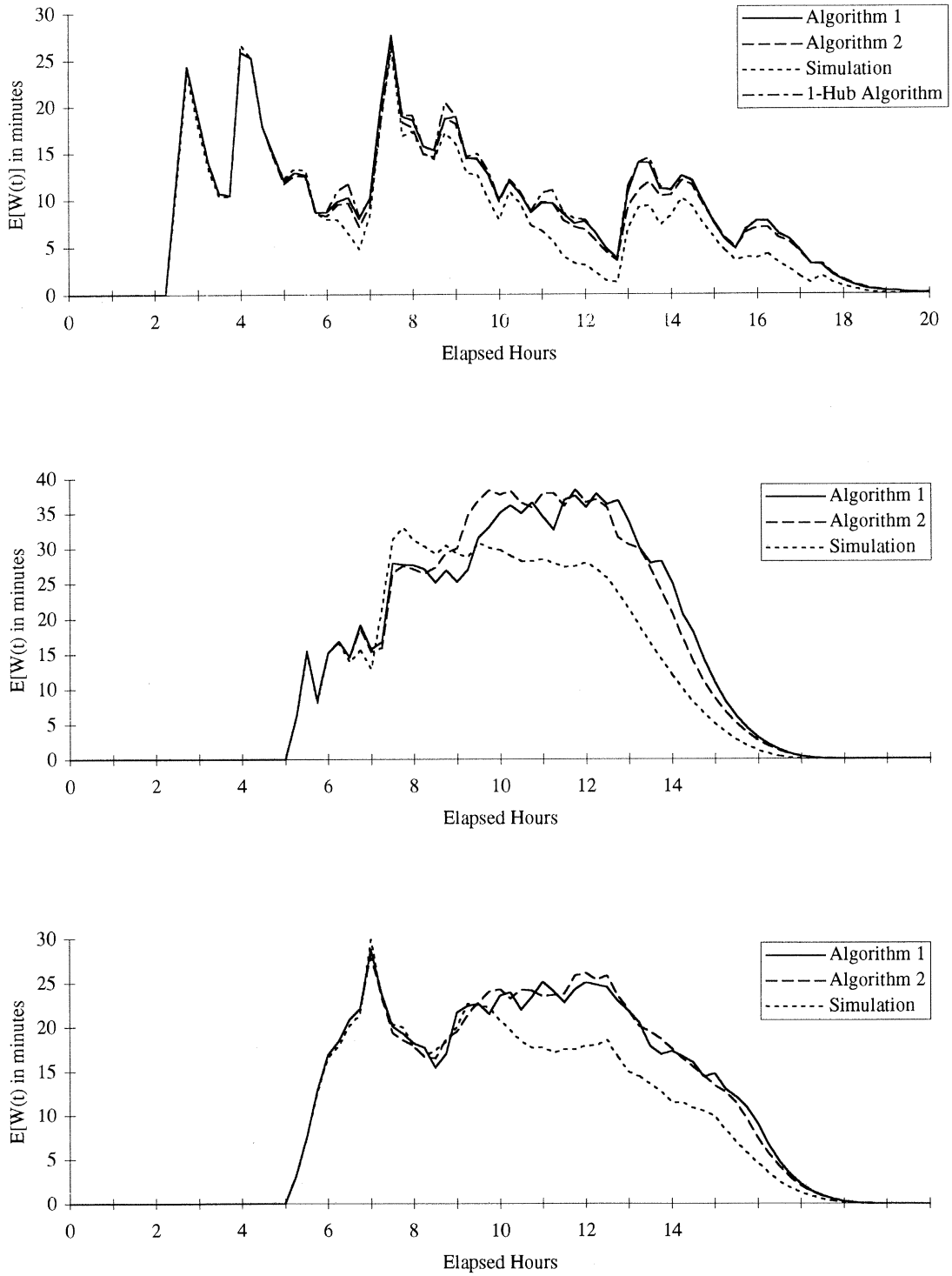

Figure 7. Comparison of expected waiting times predicted by one-hub algorithm, simulation, and the two decomposition algorithms for case 1 (DFW data) (top); comparison of expected waiting times predicted by simulation and two decomposition algorithms for case 2a (center); comparison of expected waiting times predicted by simulation and two decomposition algorithms for case 3 (bottom).

obscure the peaked arrival structure. Figure 8 plots the original demand profile at hub 1 together with that produced as a result of propagated delays (arrival rates based on averages from simulation). The figure shows quite clearly the way in which delay propagations disrupt the original peaked pattern. As noted, because of complexity limitations, the two decomposition algorithms update schedules via expected value. Because tail behavior is significant in this heavy traffic case (waiting times up to 3 hours, expected waits around 40

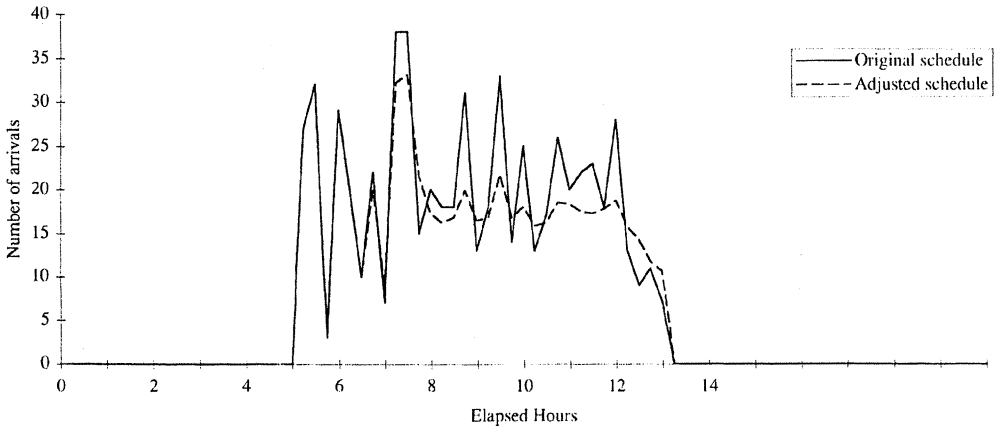

Figure 8. Influence of delay propagation on actual arrival rates by period.

minutes), the decomposition algorithms do not reflect the fact that a significant fraction of arrivals are pushed back to the later part of the day, when there is no scheduled traffic.

The results of cases 1–3 suggest the circumstances under which network impacts become important, and they also indicate that under these circumstances, the approximations developed in this paper tend to overstate waiting times during extended busy periods. Case 1 suggests that for networks of airports like DFW, waiting times are probably not high enough to create significant network effects on a frequent basis: The deterministic part of the schedule (i.e., the bank structure) predominates. Initial conditions have a major impact on schedule disruption, as the discussion in our earlier study emphasized; the low effect reported here is an average over all initial conditions which does not discount individual cases of severe disruption. In terms of average case behavior, however, high levels of propagation occur only in much heavier traffic situations where the spacing between peaks is low (cases 2 and 3). This description fits few if any airports in the nation today, though it is a plausible future scenario.

Running times for Algorithms 1 and 2 on the Ultrix workstation averaged 10 and 30 minutes, respectively. Perhaps surprisingly, the simulation procedure (10,000 replications) took only five minutes. This gap would be closed by incorporation of the restart procedure described earlier. For a value of $m = 10 - 2\ 1/2$ hours, the approximate minimum time between successive hub visits—there would be a factor 8 reduction for $K = 60$ periods, 9 for 80 periods; in other words, running times are approximately comparable. The short running time of the simulation procedure is attributable in large part to its simplicity. Like Algorithms 1 and 2, the simulation computes waiting times as if aircraft constituted a flow process; it is not a "full-fledged" simulation modeling aircraft service times as discrete events. The reason for this simplification is that the simulation's main purpose is to test the effect of the "update by expected value" approximation (the simulation updates schedules by simulated waiting time realizations rather than by expectation). A "full-fledged" simulation treating individual aircraft service times would be considerably more time intensive than all of the approaches discussed here.

Given approximately comparable running times, the main computational weakness of the two approximation procedures lies in their limited ability to reflect large delay propagations, the fact which is responsible for the discrepancies in Figure 7. For realistic demand data (case 1), the discrepancy is small; however, in heavier traffic it becomes significant. For these latter cases, the approximations probably do not model the situation as effectively as the simulation. Thus, the results are at least partly discouraging for the approximation procedures. However, they do not negate the original modeling approach for arrivals and capacity, which is incorporated into the simulation itself. Comparing this reduced-form simulation with a full-scale one is an important issue beyond the scope of the present paper.

4.2.1. A Policy Application

The models examined in the preceding subsection are useful for examining the qualitative behavior of the network under different policy scenarios concerning network *connectivity* and aircraft slack. One measure of connectivity in the test airport network is the percentage p of flights having operations at both hubs. Case 4 considers two opposing extremes of this: a fully disconnected network (case 4a), where each hub has its own set of aircraft; and a fully connected network (case 4b), where all flights alternate between the two hubs in between visits to spokes. Case 4a models the idea of hub isolation in which scheduled bank times at one hub cannot be disrupted by late arrivals from the other. It reflects a strategy in which the airline essentially operates its hubs independently of one another. In both cases, the initial capacity state of the first hub is taken to be low (poor weather), while that of the second hub is high (good weather). The phenomenon of interest is the propagation of delays from 1 to 2.

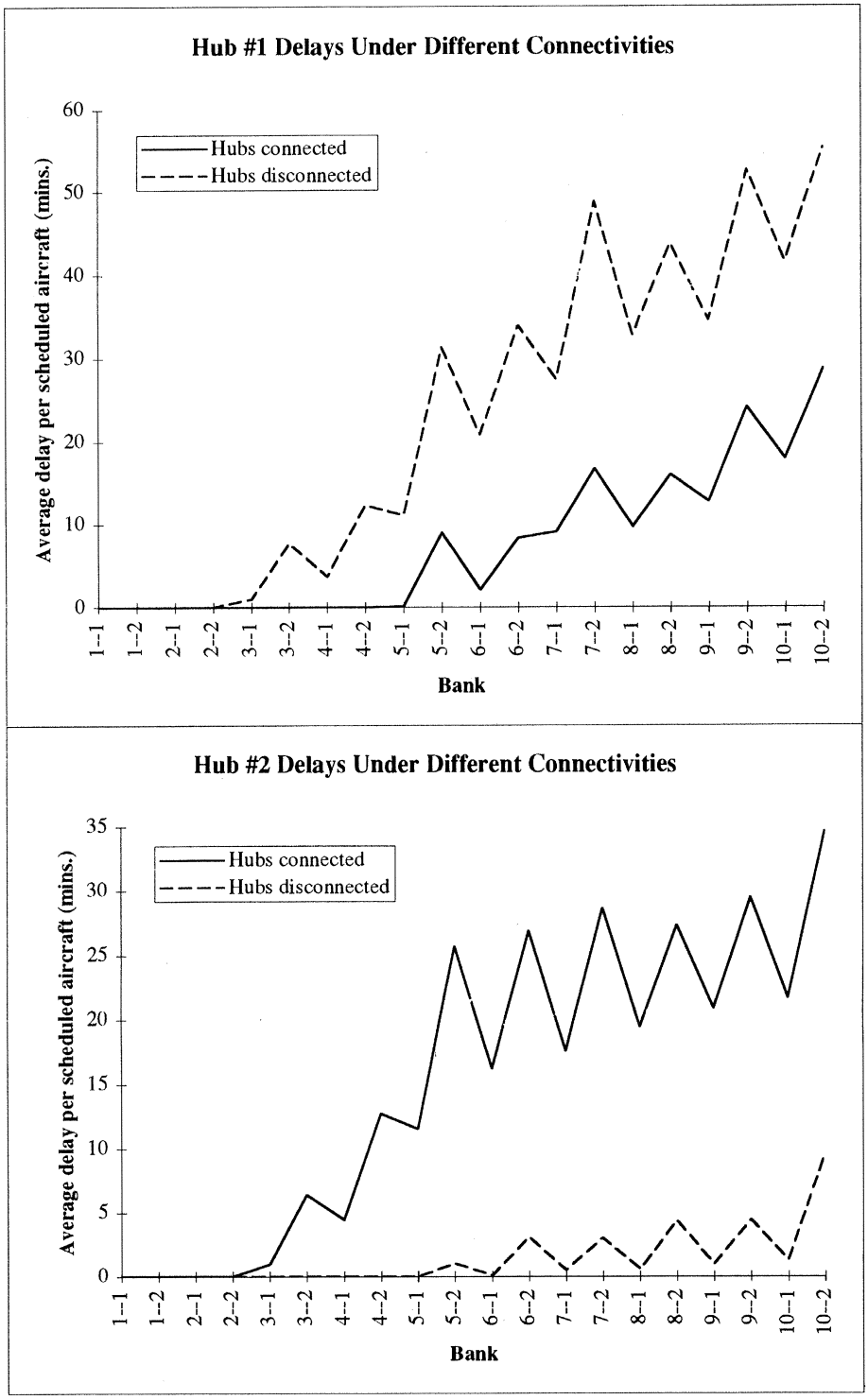

Figure 9. Average aircraft delays at two hubs under different degrees of connectivity. Note that the x-axis is in terms of banks rather than continuous time, thus 2-1 indicates the half of the second bank, 7-2 second half of the seventh bank, etc.

Figure 9 plots *average cumulative delay per arriving aircraft* (this is essentially the sum of all waiting times for the aircraft minus slack). The early banks show zero delay, while the later banks reflect delay carried over from previous points in the itinerary. The figure indicates a degradation in performance at hub 1 when it is isolated, as well as the corresponding benefits of isolation at hub 2. Conversely, the fully connected case benefits hub 1 at the expense of hub 2. The latter result is perhaps unexpected. Examining the situation more closely, one finds that the delays at hub 1 in the connected case seem to lag behind the delays in the disconnected case by about two

banks (2 hours), a circumstance explained by the fact that the minimum time between an aircraft's successive visits to the same hub is four hours in the connected case but only two in the disconnected case. This 2-hour lag does not fully explain the difference in the heights of the two curves, however. The remaining difference is explained by the fact that in the connected case, late aircraft leaving hub 1 have the opportunity of recovering some of the delay through slack at the next stop (uncongested hub 2). This opportunity is not available in the disconnected case, because the next stop is (congested) hub 1.

This result has interesting implications for a strategy of hub isolation. In the case of a hub which is believed to be the source of a large amount of congestion, such a strategy will indeed protect other hubs in the system from the uncertainties and disruptions produced by the problem hub. On the other hand, disruption at that hub itself may worsen because many of its later arrivals will have had an earlier scheduled stop there already. The carrier trades off the benefit of limiting the scope of propagation against the cost of a higher scale of delay achieved through focusing the problem in one location.

Cases 5a–d illustrate the effect of aircraft slack. As Figure 8 shows, higher slack acts to preserve the demand peaks of the original schedule and thus may actually increase local queueing delays; lower slacks smooth the schedule but do less to reduce the cumulative delay experienced by aircraft, as is illustrated in Figure 10.

5. CONCLUSION

In this paper, we have developed two related analytical models for the difficult problem of modeling transient queueing behavior in an airline network and studying the network effects of air traffic congestion. We would summarize our major findings as follows:

1. *The importance of traffic splitting phenomenon:* High uncertainty in the levels of delay encountered by aircraft is a prominent feature of the network problem. We have developed two different approximation schemes for modeling this phenomenon. When, however, successive airline banks are narrowly spaced, accuracy in keeping track of aircraft amid this uncertainty is limited by high computational complexity.
2. *Role of deterministic effects:* The peaked pattern of demand at hub airports remains a strongly determining factor in predicting waiting times, particularly when major banks are separated by adequate lengths of time.
3. *The delay and smoothing:* On the other hand, in cases where banks are narrowly spaced, delay propagation exerts a strong smoothing effect on the demand and waiting time profiles.
4. *The effects of hub isolation:* A policy of isolating a congestion-prone hub clearly does have the effect of improving performance at others. On the other hand, under this policy the isolated hub produces delays which disrupt its own future schedule.

By providing insights into such difficult issues, models of this type could serve as powerful planning tools in addressing strategic issues related to airline network design and flight scheduling. As we remarked earlier, airlines are currently undertaking efforts in this area, though from a quite different perspective and with a completely different modeling approach. The queueing approach developed here has the advantage of modeling congestion phenomena directly rather than using empirically-derived estimates of past delays. It therefore offers the ability to evaluate schedules over the range of capacity scenarios and under future traffic scenarios which are not reflected in historical data.

The decomposition approaches discussed clearly show the difficulty of the underlying queueing problem and the need for further work. Some of the difficulties are straightforward to address (e.g., the run time reductions discussed at the end of Section 1). Others, such as the adequate modeling of sample-path "splitting," are more difficult because they involve high-dimensional computational complexity. In the test problems considered in this paper, we have shown that the "tail" cases (arrivals whose delay is so large that they are pushed back to low-traffic periods) are the main source of degradation in model accuracy. Additional work is necessary to see whether this phenomenon is equally important in other test cases. In the event this proves true (as is likely), model refinements should focus on attempts at updating arrivals in a way which captures these tail phenomena more fully, without necessarily attempting to encompass the full sample space of potential demand paths. Such refinements, coupled with the inherent advantages of analytical approaches over simulation, will eventually make models such as this one a viable alternative.

APPENDIX

Network Algorithms

First Decomposition Algorithm for Air Network Congestion

Initialize

For $k = 1$ to K
  For $n = 1$ to N
    $\mathcal{A}(n, k) = \phi$
first itinerary stops are deterministic since not affected by earlier delays
For $n = 1$ to N
  For $v = 1$ to V
    $\mathcal{A}(n, \kappa(t_1^v)) = \mathcal{A}(n, \kappa(t_1^v)) \cup v$
Set $d^v = 0$ for all v.

Main Loop

For $k = 1$ to K
  For $n = 1$ to N
    Set $\lambda_k^n = |\mathcal{A}(n, k)|$.

Using the recursive method at each airport, calculate $E[W_k^1], \ldots, E[W_k^N]$.
For $v \in \mathcal{A}(n, k)$:
find the part of the itinerary corresponding to this stop
Find m: $(n_m^v, t_m^v, s_m^v) \in \mathcal{P}(v)$ and $\kappa(t_m^v + d^v) = k$.
Set $n = n_m$, $t = t_m + d^v$, $s = s_m$, $n' = n_{m+1}$, $t' = t_{m+1}$, $s' = s_{m+1}$.
Set $\alpha = \kappa(t) - t/(\Delta t)$.
calculate propagated delay
Set $d_{m+1}^v = [d^v + \alpha E[W_{\kappa(t)-1}^n] + (1 - \alpha)E[W_{\kappa(t)}^n] - s']^+$.
determine next arrival period and update data structure
Set $\mathcal{A}(n', \kappa(t' + d^v)) = \mathcal{A}(n', \kappa(t' + d^v)) \cup v$.
END.

Second Decomposition Algorithm for Air Network Congestion

Initialize
For $k = 1$ to K
For $n = 1$ to N

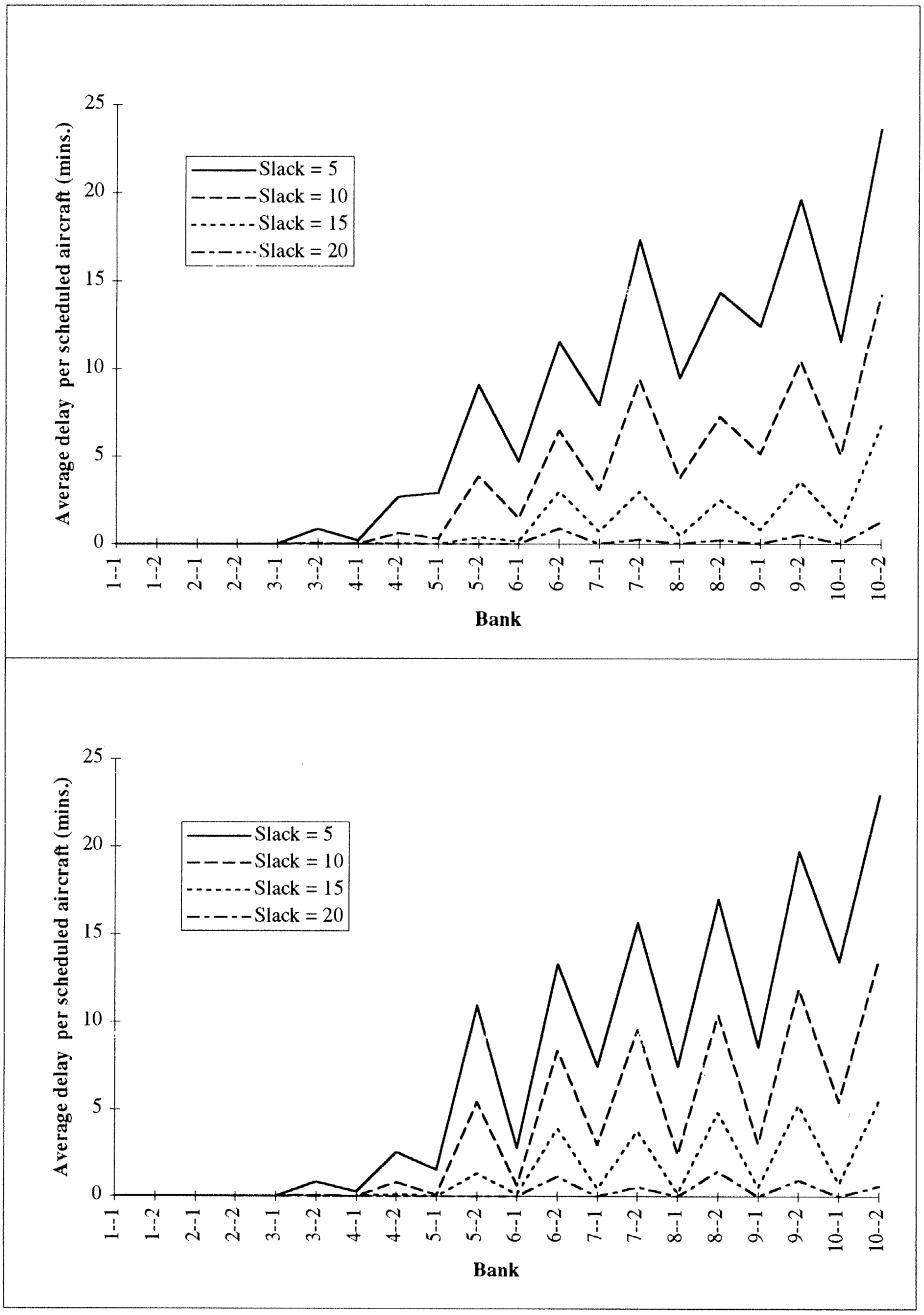

Figure 10. Effect of slack on total delay at each hub under 50% connectivity.

$\mathcal{A}(n, k) = \phi$, $E[\Lambda(n, k)] = 0$, $\sigma^2[\Lambda(n, k)] = 0$
For $v = 1$ to V
$\mathcal{A}(n, \kappa(t_1^v)) = \mathcal{A}(n, \kappa(t_1^v)) \cup v$
For each $(n, t, s) \in \mathcal{I}(v)$, $E[\lambda^n_{\kappa(t)}] = E[\lambda^n_{\kappa(t)}] + 1$
Set $d^v = 0$ for all v.

Main Loop
For $k = 1$ to K
For $n = 1$ to N
From $E[\Lambda(n, k)]$ and $\sigma^2[\Lambda(n, k)]$ determine the quantities
$\lambda^n_k(1), \ldots, \lambda^n_k(R)$ and $\gamma^n_k(1), \ldots, \gamma^n_k(R)$.
Using the recursive algorithm with probabilistic input λ, μ
calculate $E[W^1_k], \ldots, E[W^N_k]$ and $\sigma^2(W^1_k), \ldots, \sigma^2(W^N_k)$.
Update itineraries—same way as first algorithm*
For $v \in \mathcal{A}(n, k)$:
Find m: $(n^v_m, t^v_m, s^v_m) \in \mathcal{I}(v)$ and $\kappa(t^v_m + d^v) = k$
Set $n = n_m$, $t = t_m + d^v$, $s = s_m$, $n' = i_{m+1}$, $t' = t_{m+1}$, $s' = s_{m+1}$.
Set $\alpha = \kappa(t) - t/(\Delta t)$.
Set $d^v_{m+1} = [d^v + \alpha E[W^n_{\kappa(t)-1}] + (1 - \alpha)E[W^n_{\kappa(t)}] - s']^+$.
Set $\mathcal{A}(n', \kappa(t' + d^v)) = \mathcal{A}(n', \kappa(t' + d^v)) \cup v$.
Update future arrival rates
From α, $E[W^n_k]$, and $\sigma^2(W^n_k)$, determine the densities $\{f^n_k(w)\}$.
From the densities $f^n_k(w)$, determine the quantities
$p_v(0), \ldots, p_v(C)$ and $k_v(0), \ldots, k_v(C)$ for all $v \in \mathcal{A}(n, k)$.
For $c = 0$ to C:
$E[\Lambda^n_{k(v,c)}] = E[\Lambda^n_{k(v,c)}] + p_v(c)$
$\sigma^2(\Lambda^n_{k(v,c)}) = \sigma^2(\Lambda^n_{k(v,c)}) + p_v(c)(1 - p_v(c))$.

ACKNOWLEDGMENT

The work of the first author was supported by a National Science Foundation Graduate Fellowship. The work of the second author was partially supported by an NSF Presidential Young Investigator Award with a matching grant from Draper Laboratory. The work of the third author was partially supported by grants from Draper Laboratory and the Federal Aviation Administration.

REFERENCES

FITZGERALD, J. 1993. Dependability Predictor Model. Presented at the ORSA/TIMS Joint National Meeting, Phoenix, November 2, 1993.

GREEN, L., AND P. KOLESAR. 1991. The Pointwise Stationary Approximation for Queues With Nonstationary Arrivals. *Mgmt. Sci.* **37,** 84–97.

GREEN, L., AND P. KOLESAR. 1993. On the Accuracy of Simple Peak Hour Approximation for Markovian Queues. Working Paper, Graduate School of Business, Columbia University, New York.

GREEN, L., P. KOLESAR AND A. SVORONOS. 1991. Some Effects of Nonstationarity on Multiserver Markovian Queueing Systems. *Opns. Res.* **39,** 502–511.

IGLEHART, D. L., AND W. WHITT. 1970. Multiple Channel Queues in Heavy Traffic II: Sequences, Networks, and Batches. *Adv. Appl. Prob.* **2,** 355–369.

KEILSON, J., AND L. SERVI. 1990. Networks of Nonhomogeneous $M/G/\infty$ Systems. Operations Research Center Working Paper No. OR209-90, Massachusetts Institute of Technology, Cambridge, Mass.

KOBAYASHI, H. 1974. Application of the Diffusion Approximation to Queueing Networks II: Nonequilibrium Distributions and Applications to Computer Modeling. *J. Assoc. for Comput. Mach.* **21**(3), 459–469.

LAW, A. M., AND W. D. KELTON. 1991. *Simulation Modeling and Analysis*, 2nd edition. McGraw-Hill, New York.

NATIONAL TRANSPORTATION RESEARCH BOARD. 1991. Winds of Change: Domestic Air Transport Since Deregulation. Transportation Research Board National Research Council Special Report 230, Washington, D.C.

ODONI, A. R. 1991. Transportation Modeling Needs: Airports and Airspace. Volpe National Transportation Systems Center Technical Report, U.S. Department of Transportation, Cambridge, Mass.

ODONI, A. R., AND E. ROTH. 1983. An Empirical Investigation of the Transient Behavior of Stationary Queueing Systems. *Opns. Res.* **31,** 432–455.

PETERSON, M. D. 1992. Models and Algorithms for Transient Queueing Congestion in Airline Hub-and-Spoke Networks. Ph.D. Dissertation, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass.

PETERSON, M. D., D. J. BERTSIMAS AND A. R. ODONI. 1992. Models and Algorithms for Transient Queueing Congestion at a Hub Airport. Operations Research Center Working Paper No. OR272-92, Massachusetts Institute of Technology, Cambridge, Mass.

ROTH, E. 1981. An Investigation of the Transient Behavior of Stationary Queueing Systems. Ph.D. Dissertation, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Mass.

WHITT, W. 1983. The Queueing Network Analyzer. *Bell Sys. Tech. J.* **62**(9), 2779–2815.