MASTER'S THESIS

Predicting Airspace Congestion using Approximate Queueing Models

by Bala G. Chandran
Advisor: Michael O. Ball

NEXTOR MS 2002-1
(ISR MS 2002-6)

NEXTOR

National Center of Excellence in Aviation Operations Research

The National Center of Excellence in Aviation Operations Research (NEXTOR) is a joint university, industry and Federal Aviation Administration research organization. The center is supported by FAA research grant number 96-C-001.

Web site http://www.isr.umd.edu/NEXTOR/

ABSTRACT

Title of Thesis:	PREDICTING AIRSPACE CONGESTION USING APPROXIMATE QUEUEING MODELS
Degree candidate:	Bala Gautam Chandran
Degree and year:	Master of Science, 2002
Thesis directed by:	Professor Michael O. Ball R.H. Smith School of Business

We present an approximate method for the analysis of queueing delays in highly dynamic networks with schedule-based stochastic arrivals and time-varying service times. We also develop an intuitively appealing network flow model representation of the problem and compare the performance of both models to a much more detailed simulation on several sample networks. The two approaches are applied to the problem of estimating queueing delays in the airspace, which is modeled as a node-capacitated network with time-varying capacity constraints and aircraft departure-time uncertainty. We demonstrate the use of these models in airspace congestion prediction and airline schedule evaluation.

PREDICTING AIRSPACE CONGESTION USING APPROXIMATE QUEUEING MODELS

by

Bala Gautam Chandran

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland at College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2002

Advisory Committee:

Professor Michael O. Ball, Chairman/Advisor
Professor Michael C. Fu
Assistant Professor David J. Lovell

ACKNOWLEDGEMENTS

First, thanks to my advisor, Prof. Michael Ball, for his astute guidance, and unending patience and trust. Working under him was a wonderful learning experience.

Thanks to Dr. David Lovell for his advice in this research, and thoughts on life in general.

Thanks to Prof. Michael Fu for his many useful suggestions and comments.

Thanks to Bob Hoffman for being the ever-helpful benevolent big brother.

Thanks to Thomas Vossen, for helping me out on everything from code to coffee.

Thanks to Jason and Narender for making NEXTOR a fun place to work at.

And of course, thanks to Hamsa, just for being there.

This research was supported in part by the Federal Aviation Administration through NEXTOR, the National Center of Excellence for Aviation Operations Research.

TABLE OF CONTENTS

LIST OF TABLES vii

LIST OF FIGURES ix

LIST OF ALGORITHMS xii

1 Introduction 1

1.1 Motivation 1

1.2 Objectives of Study 2

1.2.1 The National Airspace System 2

1.2.2 Uncertainty in the National Airspace System 4

1.2.3 Air Traffic Management 5

1.2.4 Monitor Alert 6

1.3 Organization and Outline 9

2 Problem Definition and Literature 11

2.1 Overview 11

2.2 Problem Definition 11

2.3 Analysis of Queueing Systems 14

2.3.1 Approximate Solutions for Single Server Systems 14

2.3.2 Networks of Queues 15
2.4 Models in Air Traffic Management 18
2.4.1 Monitor Alert 18
2.4.2 The Approximate Network Delays Model (AND) 18
2.4.3 The Detailed Policy Assessment Tool (DPAT) 18
2.4.4 Other Models in Air Traffic Management 19

3 Model Description **21**

3.1 Overview 21
3.2 Modeling Philosophy 21
3.3 Stochastic Birth Schedule 22
3.4 Assumptions 23
3.5 Notation 24
3.6 Fluid Approximation 26
3.6.1 Drawbacks of the Fluid Approximation 32
3.7 Fundamental Principles 33
3.7.1 Strong interactions 35
3.7.2 Weak interactions 35
3.7.3 Sample Problem 36
3.7.4 Conditions under which Strong and Weak Interactions are Exact 37
3.7.5 Issues with interactions 39
3.8 What is a Packet? 42
3.9 Queueing Model 44
3.9.1 High-level description 44
3.9.2 Detailed Algorithm Description 44

4 Generating Occupancy Distributions **50**

4.1 Overview 50

4.2 The Occupancy Distribution 50

4.2.1 Motivation 50

4.2.2 Queue Characteristics 51

4.2.3 Notation 51

4.2.4 Problem Definition 52

4.2.5 Two Aircraft 53

4.2.6 Server Occupancy Distribution for n arrivals 54

4.3 Approximating Occupancy Distributions 56

4.3.1 The Beta Distribution 57

4.3.2 The Genetic Algorithm 60

4.3.3 Regression 64

5 Experiments **70**

5.1 Overview 70

5.2 Model Validation 70

5.3 Metric 71

5.3.1 The RCI Metric 72

5.4 Simulation 74

5.4.1 Overview 74

5.4.2 Standard Error 74

5.5 Experimental Design - Network 1 75

5.5.1 Capacity Scenarios 75

5.5.2 Drift Types 76

5.5.3 Drift Scenarios 77

5.5.4 Cancellation Scenarios 78
5.6 Experimental Setup- Network 1 78
5.7 Experimental Results - Network 1 80
5.8 Experimental Design - Network 2 86
5.9 Experimental Results - Network 2 89
5.9.1 Scenario 1 (Unconstrained) 90
5.9.2 Scenario 2 (Nominal Constraints) 91
5.9.3 Scenario 3 (Capacity Reduction at Airports) 92
5.9.4 Scenario 4 (Controls Applied to Mitigate Congestion) . . . 92
5.10 A Note on the Confidence Interval on the Model Output 93

6 Conclusions and Future Work **104**
6.1 Conclusions 104
6.2 Application 105
6.3 Other Applications 105
6.4 Future Work 106

Bibliography **112**

LIST OF TABLES

1.1 Monitor Alert Parameter (MAP) 8

4.2 Bounds placed on parameters in the genetic algorithm. 60

4.3 Algorithm parameters of the genetic algorithm. 62

4.4 Regression results for parameter α, where arrivals are uniform over the period length. 66

4.5 Regression results for parameter β, where arrivals are uniform over the period length. 67

4.6 Regression results for parameter δ, where arrivals are uniform over the period length. 68

4.7 Regression results for parameter α, where arrivals are in past periods. 68

4.8 Regression results for parameter β, where arrivals are in past periods. . 69

4.9 Regression results for parameter δ, where arrivals are in past periods. . 69

5.1 Results for hypothesis test H1: The RCI value for the model is greater than the RCI value for a fluid approximation. 82

5.2 Results for hypothesis test H2: The runtime for the model is greater than the runtime for the simulation. 83

5.3 Results for hypothesis test H3, H4, H5 : The RCI metric obtained for different cancellation scenarios is different. 83

5.4 Standard error of mean for the simulation of network 2 for scenario 2 (nominal constraints). 90

LIST OF FIGURES

1.1 Traffic Situation Display with overlays and Monitor Alert function. . . 7

2.1 Airport arrival and departure capacity tradeoff curve. 19

3.1 Modeling a stochastic schedule-based birth process. 23

3.2 A simple queueing system. 26

3.3 Network flow representation of the problem. 27

3.4 Two aircraft in an unconstrained network. 29

3.5 Probability of occupancy of the server by one aircraft. 33

3.6 Probability of occupancy of the server by two aircraft. 34

3.7 Probability of occupancy of the server by two aircraft interacting weakly. 37

3.8 Probability of occupancy of the server by two aircraft interacting strongly. 38

4.1 Sample expected occupancy distribution of a waypoint. 56

4.2 Beta distribution fit to the occupancy distribution. 63

4.3 Discrete occupancy distribution and fit beta distribution corresponding to Figure 4.2. 64

5.1 Two curves with similar shape but offset by a few periods. 72

5.2 Raw score computation of the RCI metric. 73

5.3 Test network 1. 76

5.4 Drift probability density functions. 77
5.5 Histogram of total number of scheduled departures in network 1 over time. 79
5.6 Sample output for one set of scenarios. 81
5.7 RCI values for different capacity scenarios (refer Section 5.5.1). 84
5.8 RCI values for different drift scenarios (refer Section 5.5.3). 85
5.9 Test network 2 [W - waypoint, A - airport, S - sector]. 86
5.10 Histogram of total number of scheduled departures in network 2 over time. 87
5.11 Reduced capacity scenarios for airports A-9, A-10, and A-11. 89
5.12 Cancellation probabilities for network 2 caused by reduced capacity. . 90
5.13 Reduced capacity at waypoints W1 through W8 in response to congestion. 91
5.14 Results of scenario 1 (unconstrained) for network 2 using the model. . 96
5.15 Results of scenario 1 (unconstrained) for network 2 using the fluid approximation. 97
5.16 Results of scenario 2 (nominal constraints) for network 2 using the model. 98
5.17 Results of scenario 2 (nominal constraints) for network 2 using the fluid approximation. 99
5.18 Results of scenario 3 (reduced arrival capacity) for network 2 using the model. 100
5.19 Results of scenario 3 (reduced arrival capacity) for network 2 using the fluid approximation. 101

5.20 Results of scenario 4 (controls applied in response to congestion) for network 2 using the model. 102
5.21 Results of scenario 4 (controls applied in response to congestion) for network 2 using the fluid approximation. 103

LIST OF ALGORITHMS

1 Fluid Approximation 31

2 High-level algorithm description 45

3 Detailed algorithm description 46

4 Generate Weight() 48

5 Generate Occupancy() 49

Chapter 1

Introduction

1.1 Motivation

The study of queueing networks is of interest to a wide class of researchers, due to the widespread occurrence of such networks in practice, as well as the challenges involved in studying and solving associated problems. In many real-world systems such as airports, production facilities, highways, and data networks, the costs of congestion and its propagation can be very high.

Although a vast literature exists on the analysis of queueing systems, solutions exist only for a very small set of problems, as several assumptions are required to make a network mathematically tractable to obtain exact solutions. Also, most closed-form results for queueing networks are valid only under steady-state conditions. Since many real-world systems are dynamic in nature, the applicability of exact steady-state solutions is limited. This motivates the need for other approaches to deal with complex queueing systems.

1.2 Objectives of Study

In this study we develop an approximate model to analyze the behavior of highly dynamic queueing networks with schedule-based stochastic arrivals and time-varying service times. This methodology is applied to the problem of estimating queueing delays of aircraft in the National Airspace System (NAS).

1.2.1 The National Airspace System

In this section, we describe the network structure of the NAS and introduce some terms in Air Traffic Management (ATM). The NAS, which is managed by the Federal Aviation Administration (FAA) in cooperation with the airspace users, consists of the overall environment for the safe operation of aircraft. This includes the aircraft itself, the pilots, the facilities, the tower controllers, the terminal area controllers, the enroute controllers, and the oceanic controllers. It includes the airports, the maintenance personnel and the airline dispatchers. All of this, together with computers, communications equipment, satellite navigation aids, and radars, are a part of the NAS [10].

In this study, we do not model the control and decision variables in the network; we develop the model as a purely predictive tool, which limits our definition of the airspace. Henceforth, the term "airspace" includes only the physical network and aircraft moving through them. Since the airspace essentially consists of a number of interacting entities (aircraft) moving through a system of constrained nodes (runways, waypoints[1]) that force

[1]A predetermined geographical position used for route/instrument approach definition, or progress reporting purposes, that is defined relative to a VORTAC station (navigation aid) or in terms of latitude/longitude coordinates.

queueing, it lends itself to analysis as a queueing network. Queueing is caused by capacity restrictions at nodes, driven by some minimum safety separation requirements (spatial or temporal) between aircraft. In particular, miles-in-trail restrictions (MIT)[2], determine the node capacity. For example, when an aircraft passes through a waypoint, the waypoint cannot admit any more aircraft at the same altitude for a minimum duration to ensure safety. This minimum duration is analogous to the "service time" of the waypoint. We note that this definition makes the somewhat simplistic assumption that the airspace is two-dimensional.

It is tempting to represent the NAS as a three-dimensional network of points and connecting arcs in order to model flight movements from one altitude to another. However, these altitude movements are control variables and are not known a priori as part of the flight path. Since our model assumes complete knowledge of the sequence of every server that each aircraft passes through, it is not possible to incorporate non-deterministic flight paths in our model. One possible way to deal with three-dimensional networks is to estimate aggregate capacities for flow through a three-dimensional region in space (a waypoint modeled as one server) based on traffic configurations and horizontal and vertical safety requirements. This maximum rate of flow of aircraft through a point in space translates to a service time at that point. Similarly, runways at airports are constrained by a minimum separation between aircraft at take-off and landing, which translates to a service time at

[2]A specified interval between aircraft expressed in nautical miles. Miles-in-trail is sometimes enforced as a time interval between successive aircraft, in which case it is known as a metering constraint.

the runway. It is common for miles-in-trail to be imposed at a node in order to control rates of flow into and out of a sector[3] to ensure that the number of aircraft in a sector do not reach dangerously high levels. Thus, miles-in-trail serves not only to ensure local safety, but global safety as well.

1.2.2 Uncertainty in the National Airspace System

The National Airspace System (NAS) is highly stochastic. Complete and accurate information regarding future airport and en-route airspace capacities, aircraft schedules, and aircraft flight plans are rarely available. Although several factors contribute to uncertainty in the system, unpredictability of weather is the dominant driver of randomness in the airspace. Weather uncertainty translates to error in estimating airspace capacity, which in turn leads to queueing delays and "preventive delays" as in the case of ground delay programs (readers interested in Ground Delay Programs and associated research are referred to [16]). Hence, there is considerable deviation from filed departure times, arrival times, and flight plans that cannot be estimated in advance. In this thesis, we are concerned primarily with two types of departure uncertainty:

1. Uncertainty in departure time (drift). We define drift as the degree to which the actual departure time of an aircraft deviates from its scheduled departure time in the Official Airline Guide (OAG).

2. Uncertainty in cancellation of flights.

[3]A part of airspace controlled by a team of controllers, defined, notably, by its geographical co-ordinates and its assigned radio frequency.

Idris et. al. [18] develop a detailed state-dependent model for the estimation of drift. Their research identifies runway configuration, terminal building/airline, downstream restrictions, and queue size as being causal factors in determining the taxi-out time. Although our model does not include state-dependent parameters, their study is useful in gauging the magnitude of drift, and its probability of occurrence. In our model, we could obtain drift from current methodologies used by the FAA, such as an n-day moving average for a given time of the day (not state dependent), which has been shown to have reasonable success in estimating drift.

1.2.3 Air Traffic Management

Air traffic has exhibited steady growth in the last few decades and this trend is expected to continue. However, airspace capacity has been unable to keep pace with the growth of traffic, which has resulted in increasing congestion and delays in the airspace. The directions that can be taken to avert the emerging trend toward unacceptable levels of congestion fall into four basic categories (Ball, Gosling, and Odoni [3]).

1. Capacity growth through additional airports and runways.
2. Better Air Traffic Flow Management (ATFM) at both the strategic and tactical levels.
3. Demand management at busy airports.
4. Airline operational and business strategies aimed at reducing the impact of congestion on airline schedules and costs.

Capacity growth cannot be accomplished in a short time horizon and is not an immediate solution to the airspace congestion problem. Demand management (controlling demand through pricing mechanisms) is not easily addressed either, due to regulation, and a lack of agreement in the aviation community on how best to implement such an approach. The second and fourth (tactical and operational) approaches to congestion mitigation are the most easily implemented in the near term, and require accurate and fast methods for evaluating schedules and predicting congestion in the airspace in order to react effectively through changes in schedules, routes, and miles-in-trail restrictions. This motivates the need for developing computationally inexpensive models for predicting queueing in large-scale dynamic networks under highly stochastic conditions.

1.2.4 Monitor Alert

The FAA currently uses the Monitor Alert functionality of the Enhanced Traffic Management System (ETMS) to predict congestion and alert controllers and traffic flow managers when airspace capacity is predicted to be exceeded. Monitor Alert analyzes traffic demand for all airports, sectors, and airborne reporting fixes in the continental United States, then automatically displays an alert when demand is predicted to exceed capacity in a particular area. A screen-shot of Traffic Situation Display (TSD), the user interface to Monitor Alert, is shown in Figure 1.1.

FAA regulations recommend a look-ahead period of 1.5 to 2 hours, and that action based on an alert be taken one hour prior to the alerted time frame. Traffic management initiatives are usually initiated only if the number

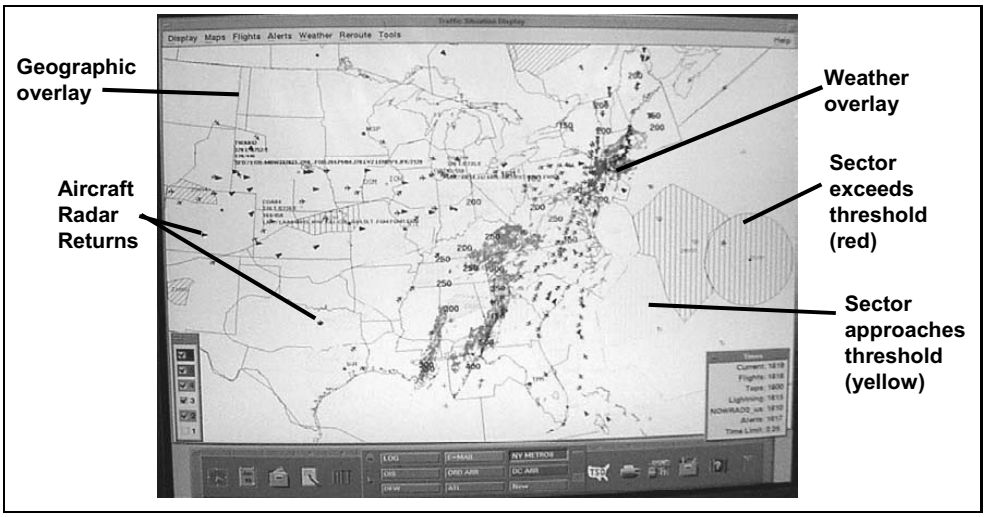


Figure 1.1: Traffic Situation Display with overlays and Monitor Alert function.

of aircraft in the sector equals or exceeds the Monitor Alert Parameter (MAP) for a sustained period of time (usually five minutes). The MAP values are listed in Table 1.1. For further operating guidelines see [11].

In order to predict the location of an aircraft over time, ETMS locates the current position of the aircraft and projects its position based on the filed speed and flight path. This prediction is adjusted to compensate for grid winds. Thus, the position of each aircraft is generated independent of other aircraft in the system. For aircraft not currently in the air, ETMS predicts positions based on the filed departure time, speed and flight path.

Consequently, Monitor Alert suffers from three major drawbacks:

- It does not account for queueing effects in the NAS caused by capacity restrictions.
- It does not account for stochastic departure times.
- It does not account for changes in flight plans.

Table 1.1: Monitor Alert Parameter (MAP)

Avg. sector flight time (min.)	MAP value (no. of aircraft)
3	5
4	7
5	8
6	10
7	12
8	13
9	15
10	17
$\geq$11	18

These drawbacks can cause significant error in the prediction. The model we propose accounts for stochastic departures and queueing effects, and is thus conceptually an improvement over the current system. We do not address the issue of changes in flight plans, as this is usually a control variable and must be considered in the context of a decision model. This thesis deals with a purely predictive model, and could later be integrated with a decision model. The proposed model serves three broad functionalities.

1. Traffic Flow Management – The model could be used to predict expected traffic flows in the network over time, based on the current schedule and current flight status. This information would then be used to set capacity restrictions on the network and to help airspace users (e.g. airlines) adjust their schedules and routes appropriately.

2. Airline schedule evaluation – The model could also enable airlines to evaluate their schedules and associated expected delays. Specifically, the model could be used to generate probability distributions for the amount of delay associated with any given aircraft in the network. Based on these delays, an airline would be able to determine the feasibility of its schedule, and make appropriate schedule/route changes.

3. Airspace capacity estimation – Utilization of the server helps measure the expected slack in the system at any given time.

1.3 Organization and Outline

The organization of this thesis is as follows.

In Chapter 2, we formally define the problem and introduce terminology that will be used. We then review results from the literature for exact and approximate methods for the analysis of queueing networks, and briefly discuss the applicability of these approaches to the defined problem. We also review existing approaches to modeling and analyzing the airspace.

In Chapter 3, we describe the modeling philosophy and components of the model. We introduce a network flow representation of the problem and then present a robust approximate model for analyzing queueing delays in highly complex network systems.

In Chapter 4, we analyze a critical component of our model (i.e., generating occupancy probability distributions of a server), and then present an approximation technique to obtain these distributions.

In Chapter 5, we describe some of the issues associated with validating a

model of the airspace and describe a continuous-time simulation model that is used to validate the approximate model. We then describe our computational experiments and present results comparing our approximate model and the flow model to the simulation.

In Chapter 6, we review the results of our analysis, and discuss the applicability of our approach to networks other than the airspace and some of the modeling aspects involved. Finally, we draw some general conclusions and discuss the scope for possible future research.

Chapter 2

Problem Definition and Literature

2.1 Overview

In this chapter, we formally define the problem and introduce basic terminology that will be used. We review some results from queueing theory for single-server systems and networks of queues. We also review some approximate methods used to analyze queueing networks and models developed specifically in the context of Air Traffic Management.

2.2 Problem Definition

We assume we are given

- An open[1] network of known topology, consisting of a finite number of servers, and a finite number of source and sink nodes. In ATM, the network would be the NAS. Airports would be the source and sink nodes. Waypoints and runways would be the servers.

[1] A system where entities are allowed to enter and leave the system.

- A birth schedule of a finite number of discrete entities for each source node in the system. We use the term "birth" to describe the event of creation of an entity at a source node, which denotes the entry of an entity into the queueing system. The event of an entity leaving the system (through a sink node) is referred to as the "death" of the entity. We prefer using this terminology to the terms "arrival" and "departure" as these terms are interpreted differently in standard queueing theory literature and Air Traffic Management. In ATM, this birth schedule is obtained from the Official Airline Guide (OAG). Each entity is a single aircraft.

- $h(i, \theta)$ – The probability density function for the departure time of each entity (the probability that an entity i is born at time θ). The deviation from the scheduled birth time can be either positive or negative. No restriction is placed on the shape of this function. In this thesis, we assume that the deviation from the scheduled birth time is strictly positive[2].

- A known (deterministic) sequence of servers that each entity in the system has to pass through before exiting the network from some sink node. This corresponds to the flight plan of each aircraft, and differs from aircraft to aircraft.

- A set of arcs between servers, with a deterministic travel time on each arc.

[2]This assumption is not restrictive; given a scheduled birth time with a known drift distribution (positive or negative drift), it is possible to obtain an equivalent scheduled"earliest birth time" with an associated drift distribution, where the drift is strictly positive.

- Service times at every server, which are time-varying but deterministic. A service time corresponds to the minimum temporal separation between successive aircraft using a waypoint/runway, that is enforced to ensure safety.
- That all servers can serve at most one entity at any given time.
- That there is no restriction on the queue length at any server.
- That entities cannot leave a queue once in the queue.
- That all queues are strictly first-in-first-out (FIFO).

The required output of the model can be phrased in many ways depending on the application.

- When the application of the model is congestion prediction: determine the expected number of entities in a given queue and/or the number of entities in transit between any two queues at any given time.
- When the application of the model is schedule evaluation: given an entity, determine the probability of death of this entity as a function of time. In air traffic, this would be the probability over time of an aircraft having arrived at its destination.
- When the application is determination of system efficiency: determine the expected slack (unused capacity) in any given server at any given time.

Answering all of these questions essentially amounts to determining the following: given an arrival profile at a server (a certain number of arrivals, each with a given arrival probability over time), compute the departure profile at

that server (probability of each entity exiting the server in each time period) for all entities at all servers. Note that the arrival profile is known from the schedule only for the first server in each sequence. The arrival profiles at all other servers are determined by the departure profiles of upstream servers.

2.3 Analysis of Queueing Systems

2.3.1 Approximate Solutions for Single Server Systems

Literature on the analysis of single server queueing systems is extensive. A fundamental assumption of most of these approaches is that the network is in steady-state, or that basic problem parameters are time-invariant. To analyze queueing at a single server under time-varying conditions, a fluid approximation model usually provides a good start (Kleinrock [21]). However, this approximation holds only when the arrival rate exceeds the service rate, and usually underestimates queueing. Diffusion approximations (Kleinrock [21] and Newell [27]) attempt to analyze queueing systems by focusing on "probability fields," rather than tangible flows of discrete entities through the system. Diffusion approximations are superior to fluid approximations since they can account for the variance in the parameters of the system. However, these approximations are also valid only under high utilization (service rate approximately equal to the arrival rate). Also, applying steady-state results to even mildly time-varying queueing systems can lead to significant error (Green, Kolesar, and Svoronos [15]).

2.3.2 Networks of Queues

Exact Results

The simplest network of queues for which closed-form solutions exist are known as Jackson networks (Jackson [19]). These systems consist of N service stations with unbounded FIFO queues, and can be either open or closed. The entities in the system are indistinguishable from each other, the input process is Poisson, and the service process is exponential with (possibly) state-dependent parameters. Jackson networks can be generalized by the so-called BCMP networks (Baskett et. al. [5]), where it is possible to have $R \geq 1$ classes of customers and service disciplines other than FIFO. The analysis of such networks is discussed in greater detail in the volume by Gelenbe and Pujolle [14].

Unlike the single-server case, there are almost no exact results for non-stationary queueing networks. Massey and Whitt [26] derive a time-dependent product form solution for an $M(t)/G/\infty$ system (a network consisting of an infinite number of queues). This infinite-server approximation is not valid in the airspace, where there are a relatively small number of constrained servers, which are sources of significant queueing.

Approximate Results

In general, the only FIFO networks with more than three queues for which the solution is known have the following characteristics (Gelenbe and Pujolle [14]).

- The network is open.
- The service time distributions are negative exponential.

- The external arrivals are Poisson.
- There is only one class of customer, or, if there is more than one class of customer, the service times are independent of customer class.
- All queues have unlimited capacity.

The number of problems that conform to this structure is extremely small. Hence, researchers resort to the following approximate methods to study complex queueing systems.

1. **Decomposition method** – This method consists of studying each network in the system by decomposing the system into a number of queues and studying each queue independently. The fundamental assumption is that the departure process from a server is a renewal process: the time interval between two departures does not depend on the preceding arrivals. This is exact in the case of Poisson arrivals and exponential service times, or when the server is saturated. Peterson, Bertsimas, and Odoni [30] apply this method to analyze queueing in the airspace.

2. **Mean value method** – The principle behind the mean value approach is that the mean response time (defined as the time between entering the queue and exiting the server) is given by the following equation.

$$E[T_i] = E[S_i] + E[S_i]E[N_i^*] \tag{2.1}$$

   Where $E[T_i]$, $E[S_i]$ and $E[N_i^*]$ are the mean response time, mean service time, and the mean number of customers in the queue respectively at the

instant of arrival. The main difficulty is to find an expression for $E[N_i^*]$. This solution approach assumes that the system is in a stationary state.

3. **Aggregation method** – The principle of this approach is to cluster servers into groups such that

   (a) interactions of variables within a group can be studied as if interactions with the exterior do not exist.

   (b) interactions of groups can be studied without the need to consider interactions between variables in each group.

   This approach involves studying each sub-system in the steady state and solving the problem with the aggregated servers using results from the steady state analysis to set effective service rates for each group. This approach would be applicable when the network naturally lends itself to decomposition. Due to the tight connectivity and strong interactions in the airspace, the two basic requirements of the decomposition approach (listed above) are usually only weakly satisfied.

4. **Isolation method** – The isolation method consists of subdividing the global system into L sub-systems and studying them separately. Ideally, the de-coupling is done in such a way that each sub-system has a known solution.

All of the above approaches require an assumption that the system is in steady state. The literature on studying approximate behavior of dynamic queueing systems is sparse due to the difficulty of the problem, as well as the fact that each approximation is usually valid only in the domain in which it was originally developed.

2.4 Models in Air Traffic Management

2.4.1 Monitor Alert

Monitor Alert is the system currently used by the FAA to predict congestion in the NAS, and is described in detail in Section 1.2.4.

2.4.2 The Approximate Network Delays Model (AND)

Malone and Odoni [25] describe an approximate model to study large-scale, weakly-connected queueing networks with stochastic and time-dependent demand and capacity and apply this model to the problem of estimating delays in the NAS. The model tries to study the queueing effects at airports and delay propagation throughout the network due to hub-and-spoke operations of airlines. Note that this problem is fundamentally different from the question we try to answer in this study; we are concerned primarily with en-route queueing and do not consider propagation of delays caused by connectivity constraints in the network.

2.4.3 The Detailed Policy Assessment Tool (DPAT)

DPAT is an air traffic simulation model developed by the MITRE Corporation's Center for Advanced Aviation System Development (CAASD). DPAT is essentially a deterministic simulation (assuming exact departure times and advance knowledge of capacities). The contribution of DPAT is not in the modeling of queueing delays, but in the modeling of trade-offs between departures and arrivals, which is critical in analyzing the propagation of delays when airport capacity is reduced. An example of such a trade-off curve is in

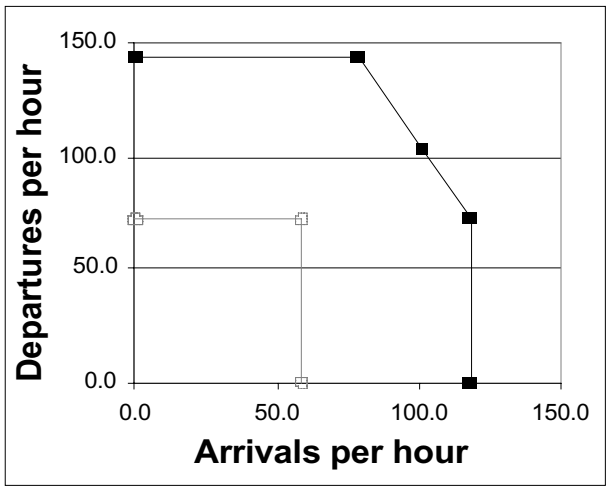


Figure 2.1: Airport arrival and departure capacity tradeoff curve.

Figure 2.1. As in the AND model, DPAT is not concerned with en-route queueing, but with airport queueing and resulting propagation effects. The reader is referred to Schaefer and Millner [33] for more details.

2.4.4 Other Models in Air Traffic Management

There are several other airspace models, including some that predict demand. Escobar [12] describes exact and approximate solutions to $M(t)/G(t)/n(t)/n(t)+q$ queueing systems and applies this to the problem Air Traffic Management, where a sector is modeled as a server with Erlangian service times. Ball, Vossen, and Hoffman [4] develop a non-stationary queueing model for the stochastic ground holding problem to determine optimal airport acceptance rates (of aircraft). Kostiuk, Lee, and Long [22] describe a forecasting model to predict long-term evolution of airline schedules in response to delays incurred. Some exact methods for the so-called aircraft-landing problem were investigated by Bell [7], Galliher and Wheeler

[13], and Pearcy [29], where the runway is modeled as a single server system with some birth process. These and other single-server models are discussed in mode detail in the volume by Saaty [32].

Chapter 3

Model Description

3.1 Overview

In this chapter, we present a robust approximate queueing model to estimate delays in non-stationary networks with stochastic schedule based arrivals. We first introduce the modeling philosophy and the procedure used to model stochastic schedule-based birth processes. We then introduce notation and present a fluid approximation to the problem and discuss some of its drawbacks. We then develop the approximate queueing model, and describe the algorithms in detail. The approximate queueing model is henceforth simply referred to as the model.

3.2 Modeling Philosophy

The basis of the model is to convert a problem in continuous time to one in discrete time by imposing convenient time buckets over the horizon of the required prediction. The duration of a "convenient" time interval (τ) depends on the application and is dealt with in more detail in Chapter 4. It is tempting

to think of the model as a discrete time queueing system; however, this would be misleading. We note that the *inputs* to the model are in *continuous time* (departure time distributions, travel times in the network, and miles-in-trail expressed as a time interval) while only the *outputs* are in *discrete time*.

Another important concept is that of "splitting" of aircraft into discrete quanta henceforth referred to as packets. It is useful to think of packets as realizations of the probability of an aircraft existing at a particular point in time. The interpretation of a packet is discussed in more detail in Section 3.8. This splitting of aircraft into packets enables us to study interactions between aircraft accounting for the probabilistic nature of the occurrence of these interactions. Throughout this study, we strongly encourage the reader to think in terms of "probability flows" rather than flows of discrete hard aircraft through the network.

3.3 Stochastic Birth Schedule

As discussed in Chapter 1, there is uncertainty regarding the time of departure of an aircraft. In order to model stochastic birth schedules, we convert the probabilities of birth at a given time to probabilities of birth during the corresponding time interval. This is simply the expected value of the function $h(i, \theta)$ (defined in the previous chapter), over the duration of the corresponding time period times the period length and is denoted by $H(i, t)$, which is the probability of birth of aircraft i during *time period* t. This is illustrated in Figure 3.1. This procedure is applied to all aircraft at all nodes. This procedure converts a departure probability distribution into a number of

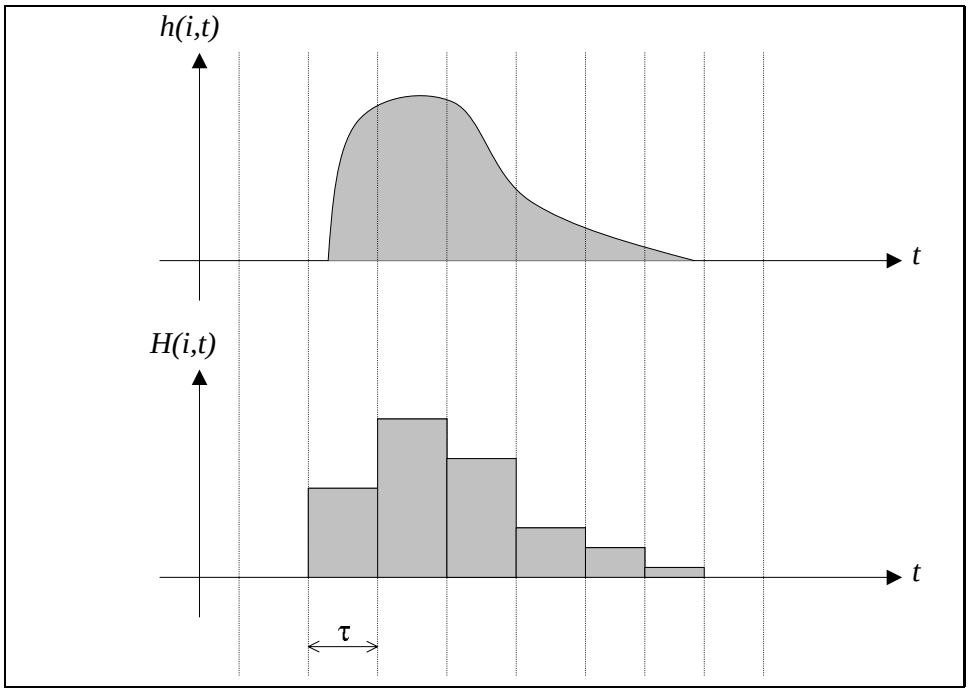


Figure 3.1: Modeling a stochastic schedule-based birth process.

departure "packets" (one in each period over the domain of the function) that are propagated through the network. This distribution can be of arbitrary shape and is usually a function of the airport and the level of queueing [18].

3.4 Assumptions

The model makes the following assumptions.

- The queue discipline is strictly FIFO.
- There are no limits on queue size / waiting time.
- The probability of birth of an aircraft during a time interval is uniform over the duration of the time interval. This is an approximation used to

convert a continuous function into a piecewise uniform function. The necessity for and implications of this assumption are discussed in Chapter 4.

3.5 Notation

The notation is developed for only a simple system of two successive queues. All the algorithms described in following sections assume that all aircraft first arrive at queue Q. Once they pass through Q, all aircraft head to queue Q'. This assumption is not restrictive as the same principles can be extended to a network of queues. Referring to Figure 3.2, an aircraft in Q is assumed to "belong" to Sector 1; once it passes through the queue, it leaves Sector 1 and then belongs to Sector 2. It is also assumed that travel time between the two queues is an integral multiple of the period length. We will demonstrate later that this assumption can be relaxed.

We now define certain important parameters and notation used.

τ	Length of a time period.
$\mu(t)$	Miles-in-trail (measured as a time interval) in the same units as the period length, as a function of the time period.
$Q(t)$	Set of all discrete aircraft quanta (packets) in the queue that arrived at the waypoint during time period t.
$Q_0(t)$	Initial state of the queue (including scheduled arrivals in future time periods).
$f(t)$	Occupancy of the waypoint in time period t.
$f_0(t)$	Initial occupancy of the waypoint.

$g^P(t)$ Occupancy of a set of packets P that are allowed to pass through the waypoint.

$C(t)$ Capacity of the waypoint in time period t.

$$C(t) = \frac{\tau}{\mu(t)}$$

$m(p)$ Mass of packet p.

$\otimes$ An operator that acts on an entire set of packets to generate new set of packets. $w \otimes Q(t)$ gives a new set of packets that have the same characteristics as the elements of set $Q(t)$, except that the masses of the packets in the new set are corresponding masses in the set $Q(t)$ multiplied by a factor w. For example, let a packet be represented as a vector (o, d, m), where o, d, and m are the origin, destination, and mass of the packet respectively. Then,

$$0.5 \bigotimes \{(a, b, 0.4), (x, y, 0.6)\} = \{(a, b, 0.2), (x, y, 0.3)\}$$

t^Q_{min} The arrival time period of the packet that has the greatest waiting time (earliest birth time) in the queue Q.

s Travel time between queue Q and the succeeding queue Q'.

$S_i(t)$ Expected number of aircraft in Sector i during time period t.

T Union of time periods in the time window being observed.

$h_i(\theta)$ The probability of birth of aircraft i at time θ.

$H_i(t)$ The probability of birth of aircraft i during time period t.

I The set of all aircraft in the NAS.

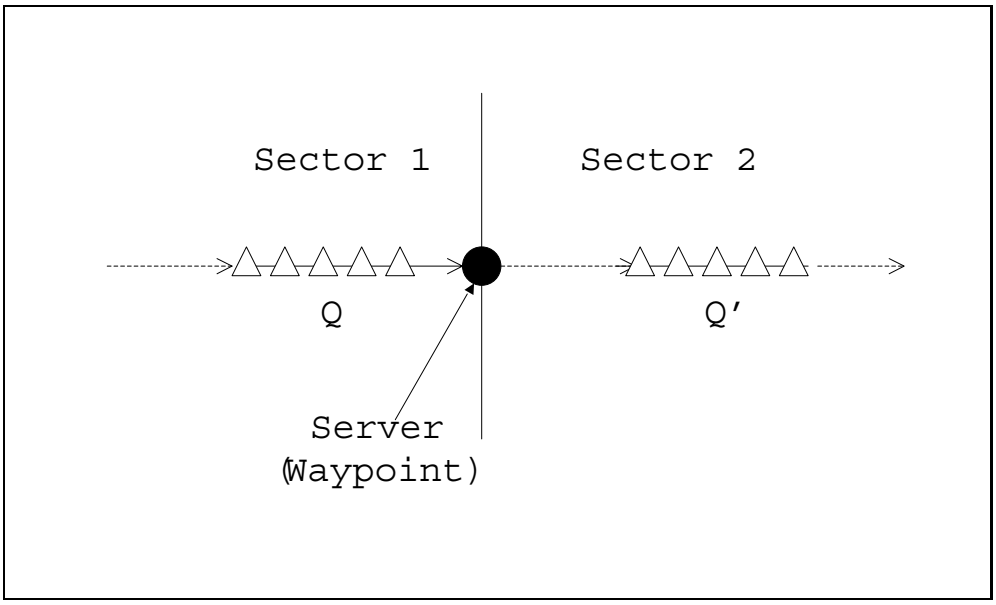


Figure 3.2: A simple queueing system.

3.6 Fluid Approximation

A first-order approach to the problem is to model the problem as a network flow model with aggregate demands and capacities (illustrated in Figure 3.3). This representation is a classic network flow model, where flow is conserved at each node. For example, the flow arrow in gray is the aggregate flow of packets that exited server i during time period $t-2$, advanced to served j, and exited server j during time period $t-1$. The delay due to queueing at server j is implicit in the time periods spanned by the flow arrow. Flow conservation at each node is applied by equating the inflow to the server at any time t to the outflow at time t. In the figure, this is illustrated for queue j at time t by the dark flow arrows. The network is node-capacitated, not arc-capacitated, where the node capacity is given by $C(t)$. The algorithm does not try to minimize any global objective function, nor does it have global constraints. Instead, it simply

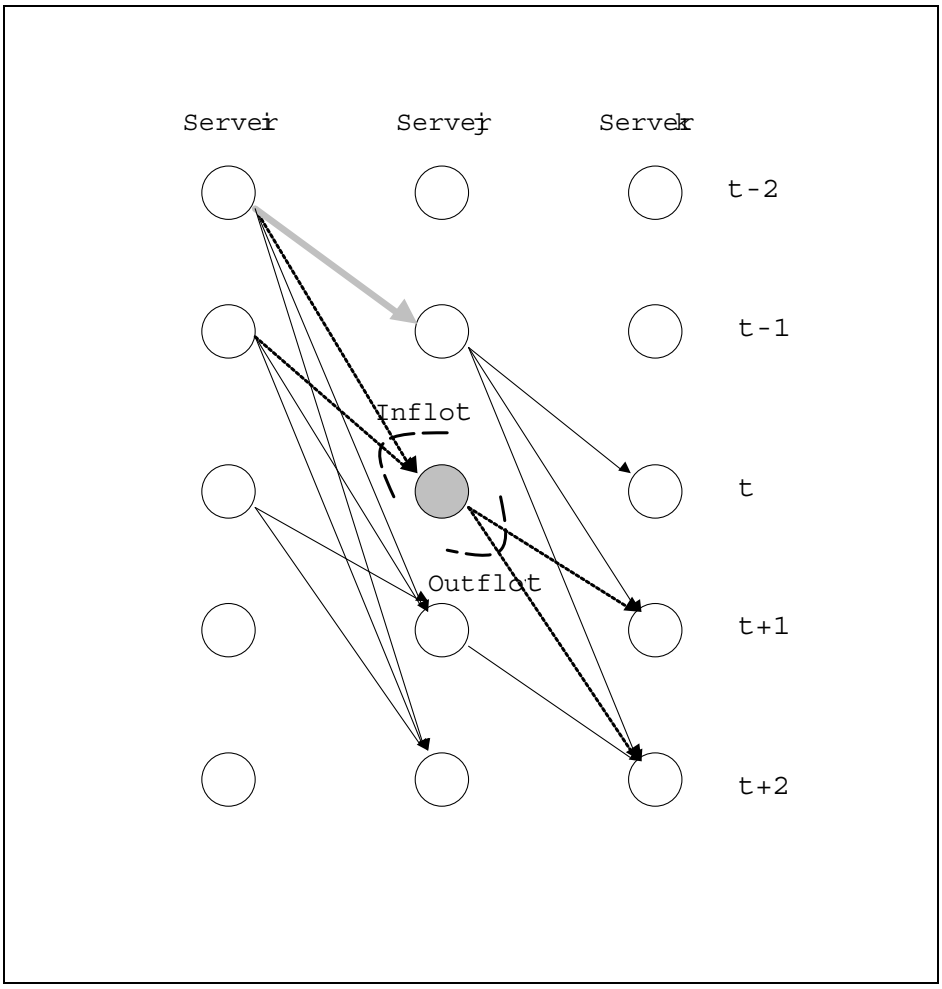


Figure 3.3: Network flow representation of the problem.

tries to maximize the utilization of each server locally, the only constraint being that flow is conserved at each server. This can be implemented using a simple algorithm that pushes a maximum amount of flow through each server (considering each server independent of all others). Thus, the algorithm simply steps through time, ensuring that the aggregate flow passing through a node during a time interval does not exceed the capacity of that node, and the

maximum possible flow is pushed through the node. In the figure,

$$Inflow(t) = Outflow(t) \leq C(t)$$

The problem thus reduces to finding a set of delay minimizing flows through the network that are capacity-feasible.

As mentioned in Chapter 2, the fluid approximation tends to significantly underestimate queueing in non-stationary queueing systems operating at less than capacity. However, this approach gives a basic understanding of the processes involved in the queueing system and some insights into modeling and implementing an aggregate queueing model, and is hence of interest. The fluid approximation algorithm is described in Algorithm 1.

Theorem 3.6.1 *When the network is unconstrained ($\mu(t) = 0 \quad \forall t$), the sector count from the fluid approximation is exactly equal to the expected sector count of the system as the length of the time period tends to zero.*

The proof follows from the linearity of the expectation and integration operators.

Proof Let $h_i(\theta)$ be the probability of departure of aircraft i at time θ. Since there is no queueing (interaction with other aircraft in the network), $\Omega_i^s(\theta)$, the probability that aircraft i is in sector s at time θ is given by

$$\Omega_i^s(\theta) = \int_{\theta-\nu_{i,max}^s}^{\theta-\nu_{i,min}^s} h_i(x)dx \tag{3.1}$$

where $\nu_{i,max}^s$ and $\nu_{i,min}^s$ are the travel times for aircraft i from the farthest point in the sector s to the departure point and the nearest point in the sector s to the departure point, respectively. This is illustrated in Figure 3.4. In words,

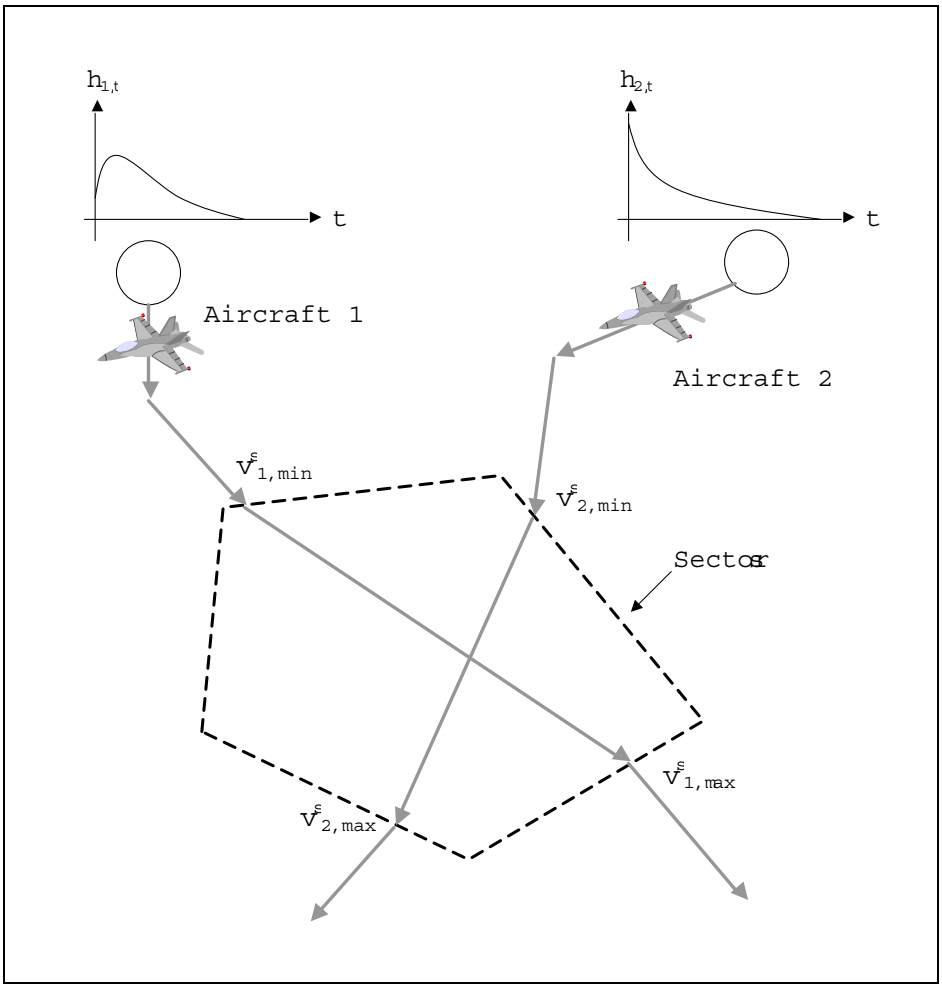


Figure 3.4: Two aircraft in an unconstrained network.

this equation states that the probability that an aircraft is in a sector at time θ is the probability that it departed in the interval $[\theta - \nu^s_{i,max}, \theta - \nu^s_{i,min}]$.

The expected number of aircraft in the sector at any given time equals the sum of the probabilities for each individual aircraft.

$$E(n, \theta) = \sum_{i \in I} \Omega^s_i(\theta) \tag{3.2}$$

Or,

$$E(n,\theta) = \sum_{i \in I} \int_{\theta - \nu^s_{i,max}}^{\theta - \nu^s_{i,min}} h_i(x)dx \tag{3.3}$$

In the fluid model, each departure "packet" with mass m is allowed to propagate deterministically through the network with an incremental change in position with each time period. Assuming that the minimum and maximum travel times to the sector are integral multiples of the period length τ, this can be stated as

$$E'(n,t) = \sum_{i \in I} \sum_{t'=t-\Upsilon^s_{i,max}}^{t-\Upsilon^s_{i,min}} H_i(t') \tag{3.4}$$

where

$$\Upsilon^s_{i,max} = \frac{\nu^s_{i,max}}{\tau}$$

and

$$\Upsilon^s_{i,min} = \frac{\nu^s_{i,min}}{\tau}$$

As defined earlier,

$$H_i(t) = \int_{\theta \in t} h_i(\theta)d\theta$$

Taking limits on Equation 3.4,

$$\lim_{\tau \to 0} E'(n,t) = \lim_{\tau \to 0} \sum_{i \in I} \sum_{t'=t-\Upsilon^s_{i,max}}^{t-\Upsilon^s_{i,min}} H_i(t') \tag{3.5}$$

Or,

$$\lim_{\tau \to 0} E'(n,\theta) = \lim_{\delta x \to 0} \sum_{i \in I} \sum_{x=\theta-\nu^s_{i,max}}^{\theta-\nu^s_{i,min}} h_i(x)\delta x \tag{3.6}$$

Which is exactly the same as Equation 3.3. ∎

Algorithm 1: Fluid Approximation

Data : Refer to Section 3.5
Result : Expected number of aircraft in a sector over time
Variable(s): M – cumulative mass of a set of aircraft;

Initialize: $f(t) := f_0(t), \forall t \in T; Q(t) := Q_0(t) \forall t \in \{-\infty, .., +\infty\}; t := 0;$

while $t \in T$ **do**

  while $(t^Q_{min} \leq t)$ *&* $(C(t) > f(t))$ **do**

    $M = \sum_{p \in Q(t_{min})} m(p);$

    if $M \leq (C(t) - f(t))$ **then**

      $w := 1;$

    else

      $w := \frac{C(t)-f(t)}{M};$

    $f(t) := f(t) + w \times M;$

    $S_1(t) := S_1(t) - w \times M;$

    $S_2(t) := S_2(t) + w \times M;$

    if $w = 1$ **then**

      $Q'(t+s) := Q'(t+s) + Q(t_{min});$

      $Q(t_{min}) := \phi;$

    else

      $Q'(t+s) := Q'(t+s) + w \otimes Q(t_{min});$

      $Q(t_{min}) := (1-w) \otimes Q(t_{min});$

  t := t+1;

3.6.1 Drawbacks of the Fluid Approximation

Experimental results obtained using the fluid approximation are presented in Chapter 5. It is observed that the fluid approximation (as expected) underestimates queueing. This is because the fluid approximation underestimates the domain of the time of occupancy of the waypoint by a set of aircraft. For example, consider the two following scenarios.

1. One entity, with a probability of birth equal to 1 in the interval [0,100] (uniformly distributed over that time interval), and a service time of 100. This generates the occupancy probability curve shown in Figure 3.5. The domain of occupancy of the server is [0,200].

2. Two entities, with probabilities of birth of 0.5 each in the interval [0,100], and a service time of 100. The probability that the server is occupied by these two entities is shown in Figure 3.6. The domain of occupancy of the server is [0,300].

The fluid approximation treats these two scenarios as being equivalent since it is concerned only with the total probability of arrival in the period (equal to 1), and not in the variance. As a result, the fluid approximation consistently underestimates the domain of occupancy of the server, and consequently underestimates queueing. This motivates the need for looking at the problem from a perspective that considers the occupancies of waypoints, and not just the flows through them.

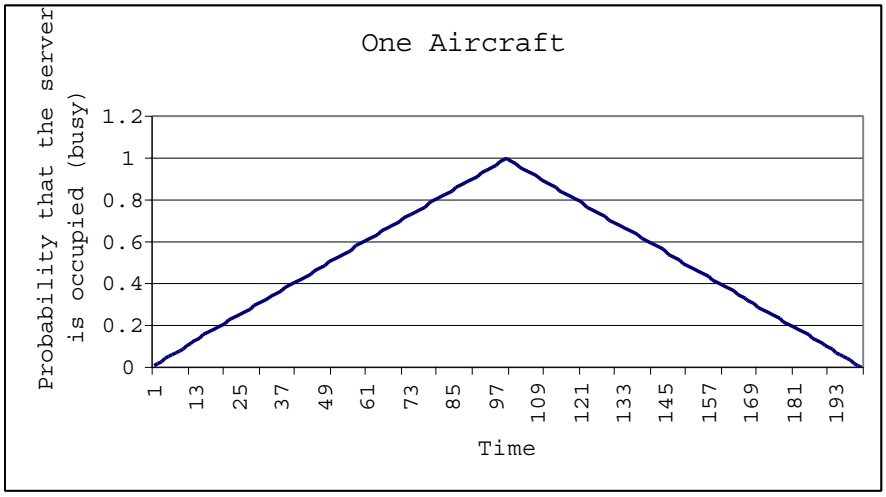


Figure 3.5: Probability of occupancy of the server by one aircraft.

3.7 Fundamental Principles

Based on our understanding of the system in the previous section, we can make some general observations regarding the mechanics of the system.

Observation 1. An entity (aircraft) passing through a server (waypoint) generates an occupancy distribution of the waypoint over time. This occupancy distribution is the probability that the waypoint is occupied by the aircraft (or set of aircraft) as a function of time.

Observation 2. An aircraft with an arrival probability density function of domain T_A which has to pass through a waypoint with an occupancy probability density distribution of domain T_O has a positive value of expected delay if $T_A \bigcap T_O \neq \phi$. This implies simply that an aircraft has to interact with the occupancy distribution of a waypoint to generate delays.

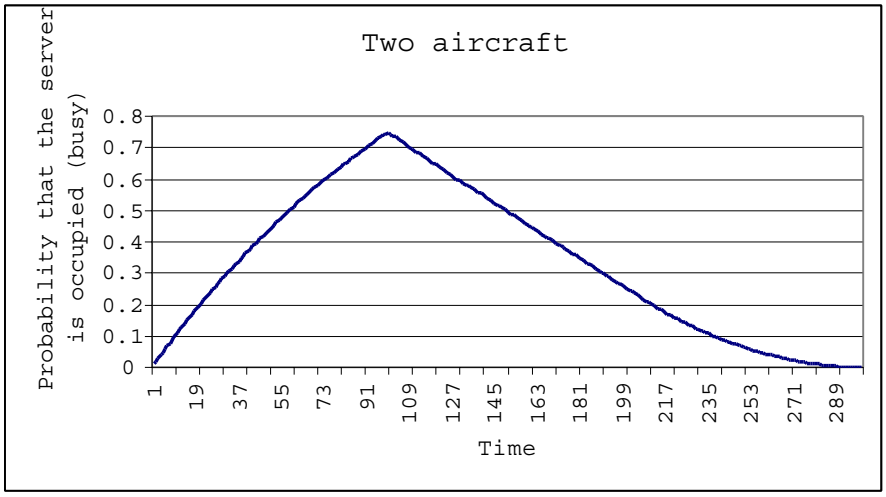


Figure 3.6: Probability of occupancy of the server by two aircraft.

If we were solving the exact case, we would require each arrival to take into account the existing probability in order to generate the correct amount of delay. In our model, since we have the possibility of an aircraft delaying itself through this procedure (two packets belonging to the same aircraft delay each other), we need another approach to estimating delays, taking the occupancy of the waypoint into account. One approach is to force delay whenever there is an overlap between the arrival and occupancy distributions, which would significantly overestimate queueing due to same-aircraft interactions. We term such interactions as strong interactions. A second approach is to recognize the presence of an occupancy probability by artificially increasing the service time at the server for the new arrivals, without explicitly causing the arrival distribution to interact with the occupancy distribution. We term this as a weak interaction. The fluid approximation is a type of a weak interaction, and underestimates queueing. In our model, given an arrival profile and an

occupancy distribution, we impose either a strong or a weak interaction depending on some conditions, which are dealt with in Section 3.7.5.

3.7.1 Strong interactions

When an aircraft arrives at a waypoint that has some positive probability of occupancy, it has a probability of being delayed. Let $p(t)$ be the probability that a waypoint is occupied at time t. The probability that the aircraft is delayed, given that it arrives at time t is given by $p(t)$. This is only the probability *that* it is delayed. Further analysis is required to characterize the length of the delay. We use the term "strong interaction" when the occupancy distribution of a waypoint interacts explicitly with the birth probability distribution to generate delays. In other words, when an aircraft arrives at a waypoint that has some probability of being occupied by another aircraft, it experiences some delay that is a direct consequence of the probability of occupancy.

3.7.2 Weak interactions

Consider two non-empty sets of aircraft (packets) arriving at a waypoint (not necessarily during the same time period). Each set of aircraft (packets) could be considered independent of each other to generate an occupancy distribution for each set. Let the occupancy probability functions of the waypoint due to each of these sets $S1$ and $S2$, considered independently, be $g^{S1}(t)$ and $g^{S2}(t)$. If these two sets of aircraft (packets) are part of a feasible flow (i.e., feasible to the queueing system), the sum these functions can never exceed 1 (since this is

a probability). If this were a feasible flow,

$$g^{S1}(t) + g^{S2}(t) \leq 1 \qquad \forall t$$

If, however the sum of these distributions were greater than 1, we would have to delay one of the sets of aircraft by some value, until the sum of occupancies of the delayed sets is not greater than 1. For example, if

$$g^{S1}(t) + g^{S2}(t) > 1$$

we would have to delay the set $S1$ by some value such that a subset of $S1$, say $S3$, passes through the waypoint such that

$$g^{S3}(t) + g^{S2}(t) \leq 1 \qquad \forall t$$

implying that $S1 \setminus S3$ is delayed. Obviously, the procedure for imposing delays to ensure feasible flows should incorporate some concept of equity between flows to ensure a first-in-first-out discipline. We use the term "weak interaction" when sets of aircraft interact with each other through their occupancy distributions to generate delays. We observe further that a fluid approximation is a form of weak interaction.

3.7.3 Sample Problem

Consider a simple system of two aircraft. Aircraft 1 has an arrival probability of 1, uniformly distributed in $[0, 1000]$. Aircraft 2 has an arrival probability of 1, uniformly distributed in $[1000, 2000]$. The miles-in-trail is 1000 units. When the two aircraft interact weakly, we simply generate the occupancies of the server independent of each other, and sum the two occupancies, ensuring that the sum is never greater than 1. This is shown in Figure 3.7. We observe that

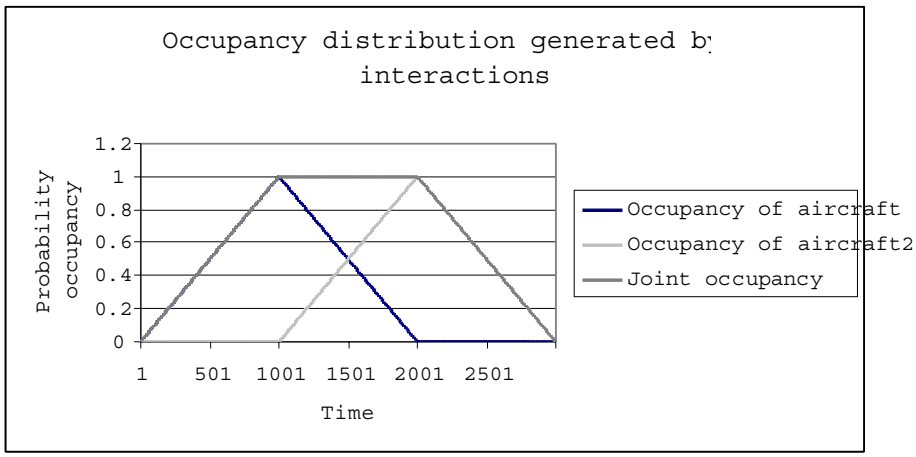


Figure 3.7: Probability of occupancy of the server by two aircraft interacting weakly.

since the sum of the two occupancies (the upper envelope in Figure 3.7) is not greater than 1, a weak interaction for this system implies that there is *no delay*.

If the two aircraft were to interact strongly (aircraft 1 explicitly delays aircraft 2), this would force some delay on aircraft 2. This is shown in Figure 3.8.

3.7.4 Conditions under which Strong and Weak Interactions are Exact

Weak Interactions

The occupancy of a server caused by a single aircraft (two packets) arriving in time $[0, 2\tau]$ (one packet in $[0, \tau]$ and one in $[\tau, 2\tau]$) with probability $f(t)$, and a

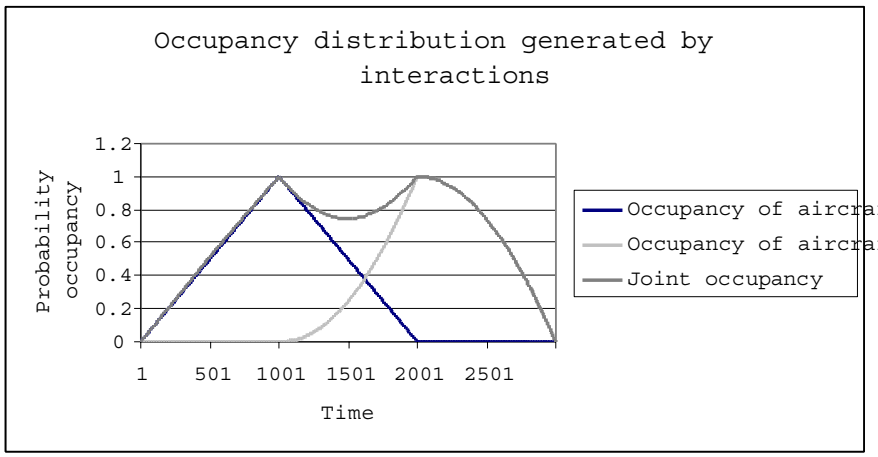


Figure 3.8: Probability of occupancy of the server by two aircraft interacting strongly.

service time of μ is given by:

$$g(t) = \int_{t-\mu}^{t} f(t)dt \tag{3.7}$$

Specifically, in $\tau \leq t \leq \tau + \mu$ (where the occupancies of the two packets overlap),

$$g(t) = \int_{t-\mu}^{\tau} f(t)dt + \int_{\tau}^{t} f(t)dt \tag{3.8}$$

The occupancy caused by a single aircraft arrival whose domain of arrival spans two periods is given by the above expression. We also observe that each of the two terms in the RHS of Equation 3.8 are nothing but the individual terms obtained for the occupancy of the server if each packet was considered in its "own" time period, i.e, $[0, \tau]$ and $[\tau, 2\tau]$, independent of the other packet.

Thus, the occupancy distribution generated by a set of packets belonging to the same aircraft (in different or in the same time period) is the sum of the individual probabilities of occupancy generated by each packet considered independently. Thus, the weak interaction is exact when all the weakly

interacting packets belong to the same aircraft.

Strong Interactions

We implement strong interactions in the model in the following manner: if an aircraft has a probability of arrival of p during some time interval, and the server has an expected occupancy of q over the time interval, the model forces a delay of at least 1 period with a probability of q. Thus, the packet of mass p splits into two packets: one of mass $(1-q) \times p$ which passes through the server with no delay, and one of mass $q \times p$ which is delayed at least one period. This procedure of assigning delay is exact if two conditions hold.

1. The length of the time period is infinitesimally small, or, all packet arrivals are at the beginning of the period (as opposed to arriving uniformly over the period duration). In our implementation, since the time period is of the order of 100 times smaller than the domain of the occupancy distribution generated by an average set of packets, this assumption holds.

2. The two interacting entities do not belong to the same packet.

3.7.5 Issues with interactions

Due to the "splitting" of an aircraft into multiple packets, it is possible to have two packets of the same aircraft interact with each other. Strong interactions would force two packets of the same aircraft to delay each other, thus overestimating queueing delays at a network. It is not possible to discount these interactions, as keeping track of occupancies for every combination of sets

of aircraft is combinatorially difficult. Weak interactions are not affected by the splitting of aircraft, as the definition of a weak interaction is such that the occupancy of a set of packets belonging to an aircraft arriving over a domain of periods is the sum of the occupancies of each aircraft in each period. Hence, using *only* strong interactions to estimate queueing would over-estimate queueing delays (an aircraft delays itself), while using *only* weak interactions would underestimate queueing (packets belonging to two distinct aircraft would not delay each other sufficiently).

To estimate queueing delays accurately, any algorithm based on these interactions would have to use a hybrid of strong and weak interactions. In our implementation of the algorithm, we make use of the fact that while strong interactions overestimate queueing, a fluid (weak) approximation underestimates queueing. Hence, the desired queueing effect may be produced by appropriately combining strong interactions with fluid interactions. From Theorem 3.6.1, we know that the fluid approximation is exact when there are no interactions between entities in the system. Hence, the fluid approximation is valid when the interactions are negligible, which occurs in three possible ways.

- The service time (miles-in-trail) is negligible.
- The total packet mass is negligible (the probability of interaction occurring is low).
- The server is overloaded (capacity is significantly exceeded), in which case entities are not given an opportunity to interact at the server.

Hence, when the above three conditions are present, we resort to a fluid

approximation, and to strong interactions in all other cases. We note that a strong interaction is also exact in the case where the system is unconstrained (miles-in-trail is negligible). This is because the occupancy generated when the miles-in-trail is negligible is also negligible, which implies that, for a strong interaction, the probability of delay is negligibly small. We prefer using the weak interaction over the strong interaction wherever possible as the weak interaction does not necessarily split a packet, while the strong interaction necessarily splits a packet to generate a new packet when the delay probability is non-zero, however small. From a computational standpoint, we would like to keep the number of packets to a minimum. As noted above, the strong interaction overestimates queueing due to same aircraft interactions. When the service times are large, the domain of the occupancy distribution of a packet is large, which increases the probability that a future packet belonging to the same aircraft interacts with itself. Specifically, when the service times are large (or when the server is overloaded), a strong interaction significantly overestimates queueing.

This leads to the issue of how to determine the threshold between strong and weak interactions. We make the following observations regarding the number of periods for which strong interactions must occur.

- If the miles-in-trail is negligible, the domain of the occupancy distribution generated by the set of packets is also negligible. Hence, as the interactions tend to zero, the domain tends to zero, and this can be approximated by a fluid flow.

- If the total quantity of packets that arrive at a server tends to zero, the interactions tend to zero. Hence, if the amount of packets remaining from

a set of arrivals is negligible, their interactions are negligible, and can be approximated by a fluid flow.

- If the number of periods that the packet has been in the queue is large, it implies that the server is overloaded. Such a situation can be approximated by a fluid flow.

From the three observations above, it is intuitive that the number of periods for which the strong fluid interactions apply must tend to zero if either the domain or the packet mass tend to zero. Hence the following relation might hold,

$$\rho = \gamma \times \text{domain} \times \text{avg.mass} \tag{3.9}$$

where ρ is the number of strongly interacting periods and γ is some positive constant. We also observe that this expression has the added property that the number of strongly interacting periods is finite. Hence, when the number of periods of delay is large, the algorithm automatically reverts to a fluid approximation. Although our experiments indicate that the value for γ appears to be sensitive to the problem parameters, a value that seems to work over a large domain of problem parameters is $\gamma = 1.5$, and this is the value that is used in all experiments. We would like to avoid calibrating the model according to the actual scenario as the model would then lose the advantage of simplicity.

$$\rho = 1.5 \times \mu(t) \times \sum_{p \in Q(t)} m(p) \tag{3.10}$$

3.8 What is a Packet?

So far we have used the term "packet" to represent some part of an aircraft that is associated with the probability of existence of the aircraft. In

particular, since the models conserve packet flows throughout, the sum of all packet masses belonging to the same aircraft equals the probability of existence of the aircraft in the system. For example, if an aircraft has a cancellation probability of 0.1, the sum of all packets belonging to the aircraft at any point in time should necessarily sum to 0.9.

The interpretation of the concept of a packet is fairly straightforward in the case of splitting the departure profile into discrete pieces of aircraft. A packet of mass m that is obtained from a continuous departure time distribution during some period is the probability that the aircraft is born (is added to the departure queue) during that period. Thus the packet is a realization of the probability of departure.

At each stage in the algorithm (each time period) it is possible to obtain the number of packets and the mass corresponding to each packet for

- Packets in transit between two queues.
- Packets in a queue.

Thus, when the algorithm has finished running, it is possible to obtain, for every arc and node in the network, a histogram (over time) of the packets belonging to the particular arc or node. This histogram is interpreted as follows: for an arc between two servers, a packet of mass m in time period t, which is a part of aircraft i, implies that there is a probability of m that aircraft i exists on the arc during time period t. Similarly, for a node (queue), a packet of mass m in time period t, which is a part of aircraft i, implies that there is a probability of m of aircraft i being in the queue during time period t.

The interpretation of a packet in the queueing process is not as obvious.

The queueing process could be viewed as a procedure that takes as inputs a set of packets, and outputs a set of packets over time with the interpretation described in the previous paragraph.

We also observe that the interpretation of a packet is strictly local. For example, we could observe the same aircraft at two nodes during the same time period (as two packets). This does not imply a finite joint probability that the aircraft is in both queues at the same time. Thus the interpretation of a packet is limited to the node/arc in which is manifests itself.

3.9 Queueing Model

3.9.1 High-level description

The intuition behind the algorithm is based on having a combination of strong and weak (fluid) interactions in the queueing systems to produce the desired queueing effect. As the algorithm steps through time, the basic idea is to generate, for each set of arrivals, the number of periods of strong interactions (from Equation 3.9). The set of arrivals strongly interacts with the occupancy distribution for the required number of periods and as a fluid for the remaining periods. The algorithm is described at a high level in Algorithm 2 and in much more detail in Algorithm 3.

3.9.2 Detailed Algorithm Description

Since it is not directly possible to generate the weight w in Algorithm 3 from the unused capacity, we use an iterative technique similar to the bisection method that converges on a weight that exactly fits the unused capacity. This

Algorithm 2: High-level algorithm description

Data : Refer Section 3.5
Result : Expected number of aircraft in a sector over time
Initialize: Queue state, Occupancy, Period = start time ;

while *Period is in time horizon* **do**

  while *Queue is not empty and occupancy is less than capacity* **do**

    Identify set of packets with earliest arrival time ;

    Determine number of non-fluid periods from domain and total mass of packets;

    if *earliest arrival time is less than (period − non-fluid Periods)* **then**

      Send as many of these packets as possible through the waypoint (not violating capacity in any period), accounting for current occupancy of the waypoint. This is a fluid approximation ;

      Update occupancies and sector counts ;

    else if *earliest arrival time is less than period* **then**

      Send as many of these packets as possible through the waypoint (not violating capacity in any period), accounting for variance in arrival time and arrival probabilities within a period and current occupancy of the waypoint. This procedure accounts for "granularity" in the arrivals, and is hence NOT a fluid approximation ;

      Update occupancies and sector counts ;

Algorithm 3: Detailed algorithm description

Data : Refer to Section 3.5
Result : Expected number of aircraft in a sector over time
Initialize: $f(t) := f_0(t) | \forall t \in T, Q(t) := Q_0(t) | \forall t \in \{-\infty, +\infty\}, t := 0$;

while $t \in T$ **do**

  while $Q(t) \neq \phi$ *&* $t^Q_{min} \leq t$ *&* $C(t) > f(t)$ **do**

    $\rho = 0.053 \times \frac{\mu(t)}{\tau} \times |Q(t^Q_{min})| \times \sum_{p \in Q(t^Q_{min})} m(p)$;

    $\chi = \frac{f(t)}{C(t)}$;

    if $t^Q_{min} < t - \rho$ **then**

      $w_{max} = 1$;

    else

      $w_{max} = \chi$;

    Generate maximum $w \in [0, w_{max}]$ such that

$$g^w \bigotimes^{Q(t)}(t') + f(t') \leq C(t') | \forall t' \in T$$

    ($w, g^w \bigotimes^{Q(t)}(t)$ are obtained from Procedure `Generate Weight` ;

    if $w = 1$ **then**

      $Q'(t+s) := Q'(t+s) + Q(t^Q_{min})$;

      $Q(t^Q_{min}) := \phi$;

    else

      $Q'(t+s) := Q'(t+s) + w \bigotimes Q(t^Q_{min})$;

      $Q(t^Q_{min}) := (1-w) \bigotimes Q(t^Q_{min})$;

    $f(t') := f(t') + g^w \bigotimes^{Q(t)}(t') | \forall t' \in T$;

    $S_1(t) := S_1(t) \sum_{p \in w \bigotimes Q(t^Q_{min})} m(p)$;

    $S_2(t) := S_2(t) \sum_{p \in w \bigotimes Q(t^Q_{min})} m(p)$;

  $t := t + 1$;

is described in Procedure `Generate Weight`. This procedure converges as long as the function is strictly non-decreasing. Since the occupancy during any time period is strictly non-decreasing with weight (increase in probability of arrival cannot cause a decrease in probability of occupancy), the procedure always converges.

Note that we use a Monte Carlo simulation in order to generate expected probabilities of occupancy as a function of time in Procedure `Generate Occupancy`. This procedure is time consuming, especially since it is performed iteratively in the Procedure `Generate Weight`. In Chapter 4, we present an approximation to the occupancy distribution, which makes it possible to obtain occupancy distributions with significantly less computation.

Procedure `Generate Weight`

Data : Current period r, Set of packets $Q(t)$, Occupancy $f(t')\forall t' \in T$

Result : Maximum weight w and expected occupancy $g^{w \bigotimes Q(t)}(t')\forall t' \in T$

Initialize: $\underline{w} := 0$, $\overline{w} := \frac{f(r)}{C(r)}$;

while $(\overline{w} - \underline{w}) > \epsilon$ *&* $\overline{w} > 0$ **do**

Estimate expected probabilities of occupancy $\underline{e(t')}$, $\overline{e(t')}|\forall t' \in T$ corresponding to $\underline{w}$ & $\overline{w}$ respectively using Procedure `Generate Occupancy`;

$\underline{g^{Q(t)}(t')} := \underline{e(t')} \times C(t')|\forall t' \in T$;

$\overline{g^{Q(t)}(t')} := \overline{e(t')} \times C(t')|\forall t' \in T$;

$\underline{\delta(t')} := \underline{g^{Q(t)}(t')} - C(t') + f(t')|\forall t' \in T$;

$\overline{\delta(t')} := \overline{g^{Q(t)}(t')} - C(t') + f(t')|\forall t' \in T$;

$\underline{\Delta} := min(0, max(\underline{\delta(t')}))$;

$\overline{\Delta} := max(0, min(\overline{\delta(t')}))$;

$w := (\overline{w}\underline{\Delta} + \underline{w}\overline{\Delta})/(\overline{\Delta} + \underline{\Delta})$;

Estimate probability of occupancy $e(t')|\forall t' \in T$ for weight w using Procedure `Generate Occupancy`;

$g^{Q(t)}(t') := e(t') \times C(t')|\forall t' \in T$;

$\delta(t') := g^{Q(t)}(t') - C(t') + f(t')|\forall t' \in T$;

$\Delta := min(0, max(\delta(t')))$;

if $\Delta < 0$ **then**

$\underline{w} := w$;

$\underline{\Delta} := \Delta$;

else

$\overline{w} := w$;

$\overline{\Delta} := \Delta$;

Procedure `Generate Occupancy`

Data : Current period r, Set of packets $Q(t)$, Weight w, Occupancy $f(t')\forall t' \in T$, Capacity $C(r)$

Result : Probability of occupancy $e(t')\forall t' \in T$

Initialize: $replications = 0$, $e(t') = 0\forall t' \in T$;

$\rho = 0.053 \times \frac{\mu(r)}{\tau} \times |Q(t)| \times \sum_{p \in Q(t)} m(p)$;

if $t < r - \rho$ **then**

$e(r) = (1 - \frac{f(r)}{C(r)}) \sum_{p \in w \bigotimes Q(t)} m(p)$;

$e(r+1) = \frac{f(r)}{C(r)} \sum_{p \in w \bigotimes Q(t)} m(p)$;

else

while $replications < N$ **do**

Randomly sample a set of arrivals from $w \bigotimes Q(t)$ based on probabilities of arrival;

Assign arrival times for these arrivals in the interval r based on probability of occupancy $\frac{f(r)}{C(r)}$;

Simulate system based on known (deterministic) arrival and service times;

Update occupancy probabilities in each period ;

$replications := replications + 1$;

$e(t')$ = average probability of occupancy over all N runs of the simulation $\forall t' \in T$;

Chapter 4

Generating Occupancy Distributions

4.1 Overview

This section describes the procedure to generate occupancy distributions for a server given a set of aircraft with probabilities of arrival during some time period. We first develop an exact solution for a simple case of two aircraft. We then extend this to an arbitrary number of aircraft for a special case. Finally, we introduce the approximation technique and describe a genetic-algorithm and regression based procedure used to estimate the parameters of this distribution.

4.2 The Occupancy Distribution

4.2.1 Motivation

The previous chapter introduced an approximate algorithm for estimating delays in complex queueing systems. The procedure hinges on being able to calculate occupancy distributions generated by a set of arrivals (packets) during a time interval. A simple approach to this is to obtain these by a Monte

Carlo simulation of the system, by sampling the arrivals from a uniform distribution defined by the packet mass, as described in Procedure `GenerateOccupancy`. However, this procedure is computationally expensive, and we would like to develop computationally less burdensome methods to estimate the occupancy distribution associated with a set of packets.

4.2.2 Queue Characteristics

The queue has the following parameters.

- Number of arrivals is finite.
- Mass of each arrival represents the probability that the aircraft arrives in a finite time interval and is uniformly distributed over the interval.
- Queue discipline is first-in-first-out (FIFO).
- Service time is deterministic but time-varying.
- Server can serve a maximum of one customer at any given time
- No limit on queue size/waiting time.

4.2.3 Notation

τ	Length of time period over which arrivals can occur.
P	Set of all aircraft $P = 1, 2, ..., n, 0 \leq n < \infty$
$\mu(t)$	Service time of an aircraft at time t.
$g(t)$	Probability of occupancy of the server at time t.
$p_i(t, t')$	Probability that *at most* i customers arrive in the period $(t, t']$.

4.2.4 Problem Definition

Given a set of arrivals uniformly distributed in $[0, \tau]$, estimate the probability of occupancy of the server as a function of $t \in [0, \infty)$. Note that we are trying to get the distribution of the expected occupancy over time, not the distribution of the occupancy at a given point in time. We would initially like to estimate the expected occupancy distribution (henceforth referred to simply as the occupancy distribution) in continuous time, before we impose discrete time slices and average over these time slices. The following basic properties of the expected occupancy distribution are known.

1. $g(0) = 0$. The server cannot be occupied before any arrivals occur.

2. $g(\tau + n\mu) = 0$. The latest time that the server can be occupied is when all n aircraft arrive at time $t = \tau$. Thus the server cannot be occupied beyond $\tau + n\mu$. In the case of time-varying drift, since the order of arrivals during a time period is immaterial, the domain of the occupancy distribution is given by $\tau + n\overline{\mu}$, where $\overline{\mu}$ is the expected miles-in-trail. The domain is calculated in practice by deterministically ordering all n aircraft, separating each by the miles-in-trail defined by the time of entry of the previous arrival, and obtaining the domain as the time of exit of the last aircraft.

3. $g(t)$ is strictly non-decreasing in $[0, \tau]$ and strictly non-increasing in $[\tau, \infty)$.

4. $g(t) < 1,\ \forall t \in (0, \tau)$ (When these are converted to expected probabilities in discrete time slices, this property would imply that the probability of occupancy in the first time period is always strictly less than one).

5. Probability of occupancy at any arbitrary time $t \in [0, \infty)$ is given by

$$g(t) = 1 - \prod_{i=0}^{\lfloor \frac{t}{\mu} \rfloor} [1 - p_i(t - i\mu - \mu, t - i\mu)]$$

   Notice that all the properties listed above follow from this property.

6. $g(t)$ can never equal 1 if all the arrivals have a mass of less than 1, since there is a finite probability that no aircraft arrive.

4.2.5 Two Aircraft

Consider the simple case of two potential arrivals at the server in the period $[0, \tau]$. The probability (mass) that customer i arrives in the period $[0, \tau]$ is $m(i)$. We assume that $\mu \leq \tau$. It is fairly straightforward to derive the equations for the case when $\mu > \tau$ using analysis similar to that presented below.

Case 1. $0 \leq t \leq \mu$

Probability of occupancy at time t is the probability that at least one (either) of the customers arrive in $[0, t]$.

$$g(t) = [1 - (1 - m(1).\frac{t}{\tau}).(1 - m(2).\frac{t}{\tau})] \tag{4.1}$$

Case 2. $\mu \leq t \leq \tau$

Probability of occupancy is the probability that at least one aircraft arrives in $[t - \mu, t]$ or the probability that both aircraft arrive in $[max(0, t - 2\mu), t - \mu]$.

$$g(t) = 1 - [1 - m(1)\frac{\mu}{\tau}][1 - m(2)\frac{\mu}{\tau}][1 - m(1)m(2)(\frac{min(t - \mu, \mu)}{\tau})^2] \tag{4.2}$$

Note that in the general case, finding this area is extremely hard due to the combinatorial complexity of the problem.

Case 3. $\tau \leq t \leq \tau + \mu$

Probability of occupancy is the probability that at least one aircraft arrives in $[t-\mu, \tau]$ or both aircraft arrive in $[t-2\mu, t-\mu]$.

$$g(t) = 1-[1-m(1)\frac{\tau+\mu-t}{\tau}][1-m(2)\frac{\tau+\mu-t}{\tau}][1-m(1)m(2)(\frac{min(t-\mu,\mu)}{\tau})^2] \tag{4.3}$$

Case 4. $\tau + \mu \leq t \leq \tau + 2\mu$

Probability of occupancy is the probability that both aircraft arrive in $[t-2\mu, \tau]$.

$$g(t) = m(1)m(2)(\frac{\tau+2\mu-t}{\tau})^2 \tag{4.4}$$

The occupancy distribution generated by such a system is shown in Figure 3.6.

4.2.6 Server Occupancy Distribution for n arrivals

It is possible to extend the results for two arrivals to n arrivals. However, it is almost impossible to enumerate all possible combinations for the general case. We will assume in developing the theory that the variance in the masses of arrivals is not significant. We assume that $m(1) \approx m(2) \approx m(3)... \approx m(n) \approx m$.

Case 1. $0 \leq t \leq \mu$

Probability of occupancy is the probability that at least one aircraft arrives in the interval $[0, \mu]$. In other words, it is (1− (probability that no customer arrives in $[0, \mu]$)).

$$g(t) = 1 - (1 - m\frac{t}{\tau})^n \tag{4.5}$$

where n is the number of arrivals with positive mass.

It can be seen that the area of this curve is easy to calculate.

$$A(0,\mu) = \int_0^{\mu} g(t).dt = \int_0^{\mu} 1 - (1 - m\frac{t}{\tau})^n.dt$$

where A is the area of the occupancy distribution in $[0,\mu]$. The arithmetic yields

$$A(0,\mu) = \mu + \frac{\tau}{m}[\frac{(1 - m\frac{\mu}{\tau})^{n+1}}{n+1} - 1] \tag{4.6}$$

Given an area under the occupancy curve of $A(t_1,t_2)$, the average occupancy in that period is calculated by averaging the area over the length of the period over which A is calculated. We obtain the occupancy in this period by multiplying this occupancy figure by the capacity.

$$g^P(t_1,t_2) = \frac{A(t_1,t_2)}{(t_2 - t_1)}.\frac{\tau}{\mu} \tag{4.7}$$

where $g^P(t_1,t_2)$ is the occupancy of the server in the time interval $[t_1,t_2]$ caused the set of arrivals P in $[0,\tau]$. It is hence possible to determine the occupancy of the server in the interval $[0,\mu]$.

Case 2. $\tau + (n-1)\mu \leq t \leq \tau + n\mu$

This is the probability that all n aircraft arrive in the time period $(\tau - t, \tau]$.

$$g(t) = (m\frac{(\tau - t)}{\tau})^n \tag{4.8}$$

Obtaining these occupancy distributions even for the case when the variance is negligible is combinatorially complex. This motivates the need for developing approximate methods to estimate occupancy distributions

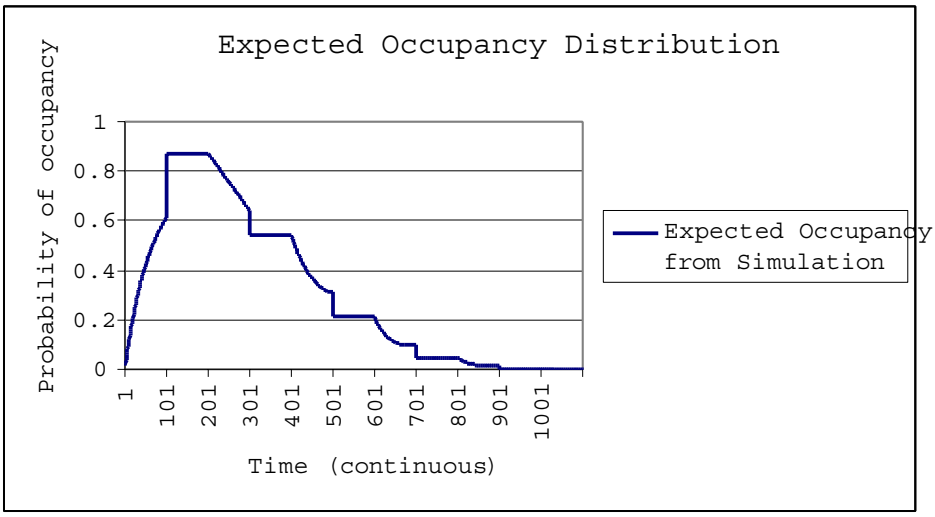


Figure 4.1: Sample expected occupancy distribution of a waypoint.

4.3 Approximating Occupancy Distributions

The success of the entire algorithm in Chapter 3 rests heavily on generating occupancy distributions quickly and accurately. As discussed in the previous section, generating exact values of the expected occupancy distribution is extremely hard due to the combinatorial nature of the computations. A look at a sample expected occupancy distribution, generated by 5 aircraft randomly arriving in [0,100] with a miles-in-trail of 200 (Figure 4.1) shows that these distributions are not very well structured. The distribution in the figure was generated based on 5 aircraft packets, which represent the arrival probabilities of the aircraft set. Using these, we first determine whether the aircraft arrives or not. Given that the aircraft arrives, the arrival time is distributed uniformly in [0,100]. This is done for all 5 aircraft and an occupancy distribution obtained for this one sample path. The entire procedure is run a number of times, and the occupancy distribution is averaged over all runs of the

replication (for each point in time) to obtain the occupancy distribution shown in Figure 4.1. The number of replications required to obtain a standard error of the mean of the order of 1% varies greatly on the number of packets being considered, and is approximately 150 replications for 50 packets. Hence, attempting to fit a smooth curve to this would only be very approximate. However, we see that due to time being discrete, we are really only interested in obtaining *areas* of the curve in different time intervals, and not the exact value of the distributions over continuous time. Hence, we would like to fit a curve to the distribution assuming that the errors are averaged out when computing areas under the curve in a time interval.

4.3.1 The Beta Distribution

This section describes the Beta Distribution, which is used to approximate server occupancy distributions. The beta distribution is a two-parameter continuous distribution related to the Gamma distribution and has two free parameters α and β. The domain of this function is $[0, 1]$, and the probability function $P(x)$ is given by

$$P(x) = \frac{(1-x)^{\beta-1}x^{\alpha-1}}{B(\alpha,\beta)} \tag{4.9}$$

Or,

$$P(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}(1-x)^{\beta-1}x^{\alpha-1} \tag{4.10}$$

where $B(\alpha, \beta)$ is the Beta function, and $\Gamma(\alpha)$ is the Gamma function. The beta distribution has the additional property that

$$\int_0^1 P(x)dx = 1 \tag{4.11}$$

The Gamma function is a generalization of the factorial function and can be applied to any real/complex number.

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \tag{4.12}$$

Note that

$$\Gamma(x+1) = x\Gamma(x)$$

For further information on the Beta distribution and Gamma function, the reader is referred to the works of Abramowitz and Stegun [1], Beyer [8], Jambunathan [20], Krysicki [23], and Zehna [35].

We choose the Beta function to approximate occupancy distributions for the following reasons.

1. The distribution is finite with domain $[0, 1]$. Since the domain of the occupancy distribution is known, it is possible to "stretch" the Beta distribution to conform to the required domain.

2. The distribution has known area (equal to 1). Given a set of packets, we know that the area of the occupancy distribution is given by

   $$A = \sum_{i \in P} m(i) \times E(\mu)$$

   where P is the set of packets, $m(i)$ is the mass of packet i, and $E(\mu)$ is the average miles-in-trail over the domain. Thus, given the area, it is possible to multiply the area of the beta distribution by the area of the occupancy distribution to obtain a good approximation to the occupancy distribution.

3. The distribution has only two parameters and has exactly one maxima. In our implementation, we introduce a third degree of freedom, Δ, which

determines the domain of the occupancy distribution.

$$D' = \Delta \times D$$

where D' is the domain of the beta distribution, while D is the domain of the occupancy distribution, given by

$$D = \tau + n \times \mu$$

4. Generating a very good approximation to the Beta distribution is computationally inexpensive.

Parameter Estimation

We use a genetic algorithm and regression based approach to estimating parameters for the Beta distribution. Given an occupancy distribution, it is difficult to use standard parameter estimation techniques to fit a beta curve to it for the following reasons.

- The beta distribution does not have structure that lends itself to a regression analysis. The derivative does not have a mathematically tractable form. Software packages that try to fit distributions to the beta distributions also rely on numerical techniques (such as regular falsi), rather than standard techniques [31]. Specifically, it is not possible to use a maximum likelihood approach since the inverse map of the cumulative distribution does not have a closed form.

- We use a three-parameter beta distribution. Such a curve has not been studied in the literature, although some methods to estimate the two-parameter distribution are known.

Table 4.2: Bounds placed on parameters in the genetic algorithm.

Parameter	Lower Bound	Upper Bound
α	1E-10	7
β	1	40
Δ	1E-10	1

- The objective is to generate a curve such that the areas formed by this curve after imposing discrete time slices on the distribution are closest to the areas formed by the occupancy distribution in discrete time. Hence, although we try to initially fit a continuous curve to the continuous occupancy distribution, the objective of the fit is to find a curve that minimizes the deviation of the discrete beta distribution from the discrete occupancy distribution.

4.3.2 The Genetic Algorithm

This section describes the process of estimating the parameters of a Beta distribution corresponding to a specific occupancy curve i.e. given an occupancy distribution, it obtains the "best fit" Beta distribution for that curve. The notion of a "best fit" is discussed later in this section. A genetic algorithm is used to estimate these parameters, and is fairly simplistic. The GA has the following characteristics.

Representation – The chromosome is a vector of three parameters - α, β, and Δ. Bounds are placed on the values of these values to focus the search. These bounds are listed in Table 4.2.

Operators – The following operations are performed on the chromosomes.

- Partial mutation – A single parent chromosome is perturbed within the bounds to generate a new chromosome in the neighborhood of the parent.
- Full mutation – A chromosome is generated randomly within the bounds. This ensures diversity in the population.
- Crossover – The arithmetic mean of the two best chromosomes vectors is obtained to form the child chromosome. Care should be taken to see that the two parents being combined are not essentially the same, as this would produce a child that is the same as the two parents. In order to control for this, we allow a crossover only when the Euclidean distance between two parents is greater than some minimum threshold. If the distance between the two best chromosomes is less that the threshold, the first and third chromosomes are considered for the crossover and so on. If the distance from the best parent to all the other chromosomes is less than the threshold, the farthest chromosome in the population from the best chromosome is chosen for the crossover.

Fitness Function – As mentioned earlier, we are trying to fit a curve to the continuous occupancy distribution, although our real objective is to fit a discrete curve to the discrete occupancy distribution. Hence, our fitness function is to minimize some linear combination of the squared deviation between the continuous curves and the squared deviation between the discrete curves. We retain the continuous curve in the objective as it is sometimes possible to get an exact fit on the discrete curve, while having large deviation

Table 4.3: Algorithm parameters of the genetic algorithm.

Parameter	Value
Crossover probability	0.6
Partial mutation probability	0.15
Population Size	10
Number of generations	$\lfloor \sqrt{100 \times n \times \mu} \rfloor$
Weight for discrete fit in fitness fn.	0.7
Weight for continuous fit in fitness fn.	0.3
Number of replications in the simulation	10000

in the continuous curve. This could lead to significant instability in the parameter estimation of two similar curves (a small change in the problem parameters should not give rise to a large change in the parameters of the beta distribution).

All the parameters of the GA are listed in Table 4.3. Since the search space is small in the GA used, diversity in the initial population ensures that at least some members of the population lie close to the required solution. Hence, the value added by the mutations is marginal, and we would like the crossover to dominate the GA (giving it a probability of occurrence of 0.6). In order to search through sufficient solutions, we could either increase the number of generations or the size of the initial population arbitrarily. A large number of generations forces the population to converge to a single parameter, while increasing the population size causes greater initial diversity in the population. In our implementation, we choose an initial population size of 10,

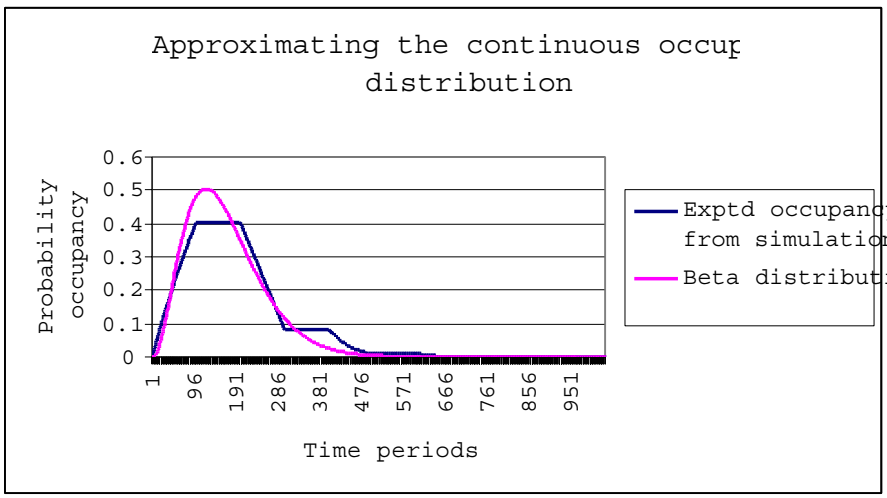


Figure 4.2: Beta distribution fit to the occupancy distribution.

since we have only three parameters, which ensures sufficient diversity in the initial population. The initial population is sampled randomly between the bounds in Table 4.2. We observe from computation that the greater the domain of the curve of the distribution($n\mu$, the greater is the number of generations required for convergence. Again, from preliminary computation, we observe that the number of required generations should be a concave function of the domain. The number of replications of the simulation used to generate the occupancy curve is much larger than would be necessary to obtain a reasonable standard error. However, we run an arbitrarily high number of replications so that the distribution obtained is very nearly exact.

Sample curves for a beta distribution fit to an occupancy curve are shown in Figures 4.2 and 4.3 (the parameters for the distribution are $\alpha = 3.27292$, $\beta = 33.5801$, and $\Delta = 0.886337$).

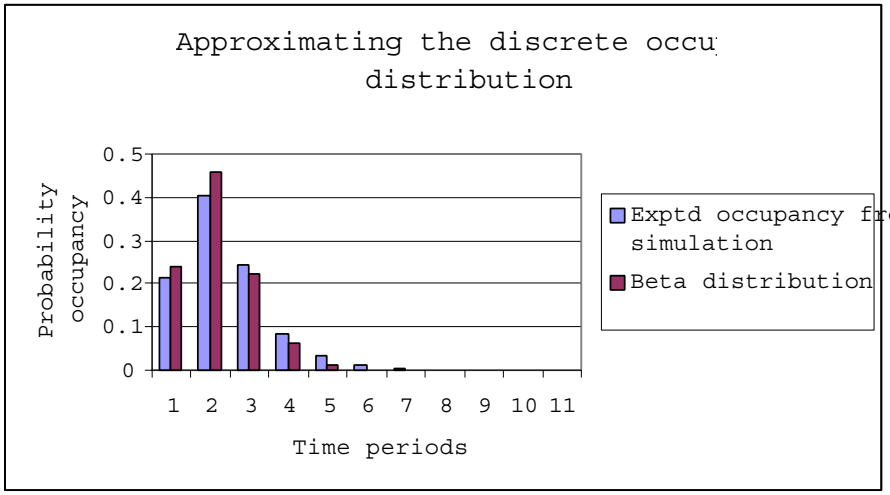


Figure 4.3: Discrete occupancy distribution and fit beta distribution corresponding to Figure 4.2.

4.3.3 Regression

The genetic algorithm is applied to 1000 instances of the problem varying number of packets, miles-in-trail, packet mean, and packet standard deviation, and the parameters obtained for each of the 1000 cases. We then run a stepwise regression each for α, β, and Δ against the problem parameters to obtain the final estimated parameters. Stepwise regression is a regression technique where the "best" regression equation is obtained by selecting a subset of the possible explanatory variables. The choice of whether or not an explanatory variable is used in the regression equation depends on the p-value associated with the variable. The question the stepwise regression attempts to answer is : what is the added value in predictive power obtained by each variable in the presence of other variables. Thus, if two explanatory variables were highly correlated, and the independent variable could be predicted using only one of the

variables, the stepwise regression would used only one of the variables (the one with the higher p-value). In this thesis, we run three stepwise regressions (one each for α, β, and δ as the independent variables). The explanatory variables used were: the number of packets (n), the mean packet mass ($\overline{m}$), the standard deviation in the packet mass (σ), the coefficient of variance $\frac{\sigma}{\overline{m}}$), total mass ($\overline{m} \times n$), the miles-in-trail as a fraction of the period length ($\frac{\mu}{\tau}$), and the range in units of the period length ($\frac{n\mu}{\tau}$). These explanatory variables capture all the factors that intuitively affect each of the three parameters.

We run two sets of regressions: one set where the arrivals are in the current period (uniformly distributed over the period length), and one set where the arrivals were in past periods, and hence arrival times are all at the beginning of the period. There were thus 6 regressions in all (two sets, three parameters each). The p-value threshold for the entering variable was set to 0.05, and the threshold for the leaving variable in the stepwise regression was set to 0.1.

Regression Results

Arrivals in Current Period.

The results for the regressions and the corresponding coefficients for α, β, and γ are shown in Tables 4.4 through 4.6.

Arrivals in Past Periods.

The results for the regressions and the corresponding coefficients for α, β, and γ are shown in Tables 4.7 through 4.9.

The following equations were obtained using the linear stepwise regression. We need two sets of regressions: one set where the arrivals are in

Table 4.4: Regression results for parameter α, where arrivals are uniform over the period length.

Summary measures			
Multiple R	0.5653		
R-Square	0.3196		
Adj R-Square	0.3104		
StErr of Est	0.4194		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	2.2160823345	0.0930	0.0000
$n\overline{m}$	-0.1331977546	0.0303	0.0000
$n\mu/\tau$	-0.0029349441	0.0008	0.0003
$\overline{m}$	-2.8703272343	1.2905	0.0271

the current period (to account for variance in the arrival period), and one set where the arrivals are in past periods (all arrivals are present at the beginning of the period).

Interestingly, the standard deviation in the masses of the arriving packets was not significant in any of our tests. This is because the standard deviation makes a difference only when the number of packets is extremely small. Given typical airspace traffic operating around capacity with drift uncertainty of the order of tens of minutes, the traffic is such that there is of the order of 50-100 packets in each arrival set. Hence, the standard deviation does not help significantly in predicting the occupancy distribution. It should be noted that

Table 4.5: Regression results for parameter β, where arrivals are uniform over the period length.

Summary measures			
Multiple R	0.6234		
R-Square	0.3886		
Adj R-Square	0.3831		
StErr of Est	7.9278		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	28.1885032654	1.4607	0.0000
$\overline{m}$	-213.0214691162	20.5559	0.0000
$n\overline{m}$	3.0105283260	0.4185	0.0271

the regression equations will change based on the application, and hence the model should be re-calibrated according to the situation. Although it may appear that the second set of equations does not take into account the miles-in-trail (none of the parameters are functions of the miles-in-trail), the miles-in-trail is implicitly considered in the domain of the distribution.

Thus, given a set of arrivals in the current period or past it is possible to generate the corresponding occupancy distribution extremely quickly and accurately.

Table 4.6: Regression results for parameter δ, where arrivals are uniform over the period length.

Summary measures			
Multiple R	0.2887		
R-Square	0.0834		
Adj R-Square	0.0793		
StErr of Est	0.2183		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	0.6086220741	0.0369	0.0000
$\overline{m}$	2.5317628384	0.5609	0.0000

Table 4.7: Regression results for parameter α, where arrivals are in past periods.

Summary measures			
Multiple R	0.5382		
R-Square	0.2897		
Adj R-Square	0.2863		
StErr of Est	0.4132		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	1.9834302664	0.0476	0.0000
$n\overline{m}$	-0.1926418394	0.0210	0.0000

Table 4.8: Regression results for parameter β, where arrivals are in past periods.

Summary measures			
Multiple R	0.6699		
R-Square	0.4488		
Adj R-Square	0.4435		
StErr of Est	8.0627		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	21.8554306030	1.7470	0.0000
n	0.1729187816	0.0227	0.0000
$\overline{m}$	-118.5249710083	18.6612	0.0000

Table 4.9: Regression results for parameter δ, where arrivals are in past periods.

Summary measures			
Multiple R	0.3969		
R-Square	0.1575		
Adj R-Square	0.1494		
StErr of Est	0.2168		
Regression coefficients			
Variable	Coefficient	Std Error	p-value
Constant	0.8020090461	0.0260	0.0000
$n\overline{m}$	0.0777332187	0.0166	0.0000
n	-0.0051935352	0.0008	0.0000

Chapter 5

Experiments

5.1 Overview

In this chapter we describe our computational experiments and present our results. We also discuss some of the issues associated with validating our model. We adapt an existing metric to our problem and show how it is used to compare the performance of our model to the simulation.

5.2 Model Validation

The primary objective of the model is to predict congestion, usually unacceptably high levels of congestion in the airspace. If the outputs of the model were acceptable (sector thresholds are never exceeded) our model would be redundant, as there would be little motivation to change flight plans or schedules. Hence, we are essentially trying to predict situations where the capacity of the airspace is exceeded, in order that the controllers and planners can make effective changes. In reality, when such a situation threatening safety occurs, some control is applied to prevent such an occurrence making it is

almost impossible to validate our model against 'real" data, since a situation that our model tries to predict is never allowed to occur. There are two approaches that can be followed to validating a congestion prediction model (Voss and Hoffman [34]).

- Deduce the presence of congestion from historical records of control applied.
- Perform a simulation of the airspace.

This problem has also been studied by Beaton et. al. [6] using a combination of deduction of congestion and a simulation. In our study, we use a simulation of the airspace to validate our model, since we would like to validate it against a very large range of test problems.

5.3 Metric

In order to compare the output of our model to the results of the simulation, we essentially need to compare two discrete distributions against each other (number of aircraft in a sector against time). Note that in this study, we are comparing the expected value from the simulation against the model output. We choose to compare the number of aircraft because comparing predictions of congestion would depend on the definition of congestion, which is subjective.

It is tempting to compute the error as a sum of absolute or squared deviations, but this could give rise to a significant error value when the shapes of the two distributions being considered are similar but are offset as in Figure 5.1.

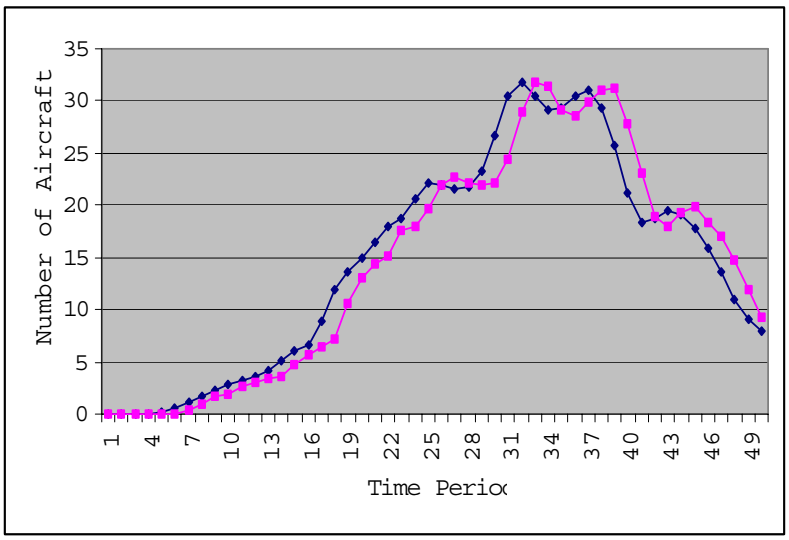


Figure 5.1: Two curves with similar shape but offset by a few periods.

We would hence need a metric that does not penalize an offset in the distributions as much as it penalizes a fundamental difference in the shapes of the distributions. Such a metric, called the Rate Control Index (RCI) metric, was proposed by Hoffman and Ball [17] in a different context, where the objective is to compare achieved traffic flow to targeted traffic flow. This methodology can be applied to any comparison of two discrete functions of time. We apply the aggregate version of the RCI to the problem of comparing sector counts obtained from the model to those obtained from the simulation.

5.3.1 The RCI Metric

The RCI measures the flow of air traffic into an airport prior to any airborne holding that may occur and compares it to the targeted flow. The aggregate version of the RCI involves using a greedy algorithm to compute the number of

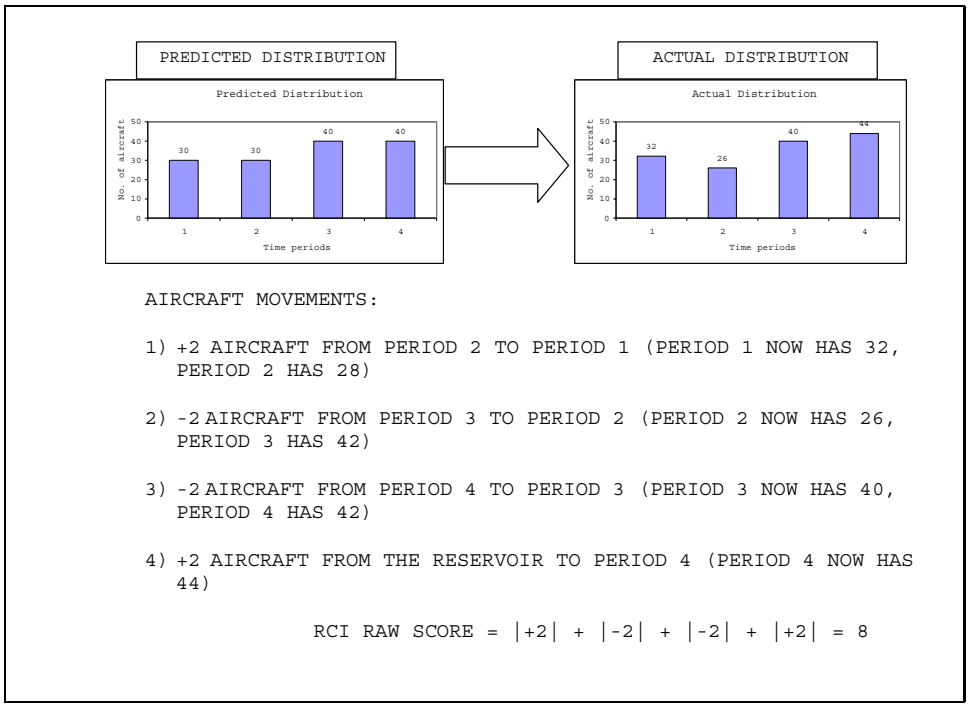


Figure 5.2: Raw score computation of the RCI metric.

"moves" that must be made in the realized distribution so that it equals the planned distribution. This is then normalized by the cost of a worst case distribution. The greedy algorithm is illustrated in Figure 5.2. The RCI metric computes the (fictitious) flight movements required to transform the realized traffic distribution into the planned distribution. The aggregate version of the RCI tends to penalize errors farther away from the reservoir (in the example in Figure 5.2 the reservoir is at the "end", which would penalize errors early on more.). In our implementation, we would like not to penalize errors for the time of occurrence. Hence, we run the greedy algorithm twice (with the reservoir in the "beginning" and at the "end"), and average over both cases. We finally normalize by dividing the RCI score by the area of the actual distribution.

5.4 Simulation

5.4.1 Overview

We use a continuous time simulation to validate our model. The simulation is a simple Monte Carlo simulation which samples aircraft departures from the given drift distribution and deterministically propagates these aircraft through the network, ensuring FIFO at all queues. The number of replications of this simulation depends on the network size and uncertainty in the system (drift). The reader interested in the actual implementation and code is referred to [9].

5.4.2 Standard Error

The standard error of the mean is defined as

$$\sigma_M = \frac{\sigma}{\sqrt{N}}$$

where σ is the standard deviation of the observations, and N is the number of observations that the distribution is based on. In our case, since we have a distribution for the number of aircraft in a sector during each time interval, we have a standard error for each time interval for each scenario for each sector. We do not report all the standard errors. Instead, we report the standard errors for only a typical case, where the standard error is defined as

$$\sigma_M = \frac{\overline{\sigma}}{\sqrt{N}}$$

where $\overline{\sigma}$ is the standard deviation averaged over all time periods.

5.5 Experimental Design - Network 1

We run our algorithm and compare it to the simulation and a simple fluid approximation on the network shown in Figure 5.3. We test our algorithm extensively on this network as it has simple structure, and it easy to detect errors, if any, using this network. If the model works well on this network, by predicting the expected outflow from the queue accurately, it follows that the model will work on larger networks as well, as a larger network is a number of smaller networks in combination. We believe that the performance of the algorithm will improve as the network gets more complex, since greater traffic leads to better averaging of stochastic variances in the system. This network consists of two arrival and two departure airports and all aircraft flow from Sector 1 to Sector 2 (refer Figure 5.3). Queueing occurs at the waypoint, and, in some cases, at the airports. We study both cases, as the drift inputs to the model could either be the deviation in take-off time of the aircraft from the scheduled take-off time (airport queueing implicitly considered) or the deviation of the gate push-back time from the scheduled push-back time (airport queueing has to be explicitly imposed).

5.5.1 Capacity Scenarios

We run a number of capacity scenarios, with increasing "difficulty" to track the performance of the algorithm as we move from one scenario to another. The scenarios used are:

1. Unconstrained. Due to Theorem 3.6.1, we expect to model to give near-exact results.

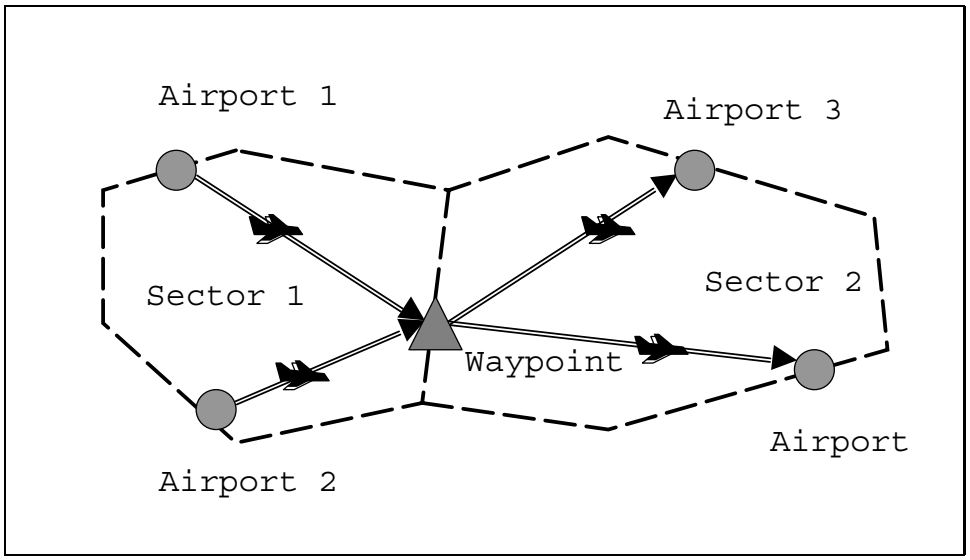


Figure 5.3: Test network 1.

2. Low constant (stationary) constraints.

3. Medium constant constraints.

4. High constant constraints.

5. Time varying capacity (medium constraints) with low variance.

6. Time varying capacity (medium constraints) with high variance.

5.5.2 Drift Types

As mentioned previously, we run the models for two types of drifts: one that implicitly considers airport queueing, and one that needs airport queueing to be explicitly modeled. When the airport queueing is implicitly accounted for in the drift, the drift is triangular shaped as shown in Figure 5.4(a). The actual shape of the drift was not experimentally determined. We use this distribution

since it is a very simple finite distribution, and does not make any special assumptions that would enable the problem to be solved more easily than any other distribution. The maximum drift tends to be of the order of one hour in this case, as this drift includes delays incurred on the runway, as well as delay in push-back. The second type of drift is when the queueing on the runway has to be explicitly modeled. The drift data provided gives the deviation of the gate push-back time from the scheduled time. This drift distribution is also triangular shaped, but the mode is at the scheduled time of push-back. This makes intuitive sense, as we would expect to see some finite non-increasing function over time. Again, the triangular shape has not been experimentally validated, and is used in this study as a simple and intuitive function to model drift. This distribution is shown in Figure 5.4(b). The maximum drift is allowed to vary over a large set of values (10 minutes to 1 hour), in order to test the robustness of the model.

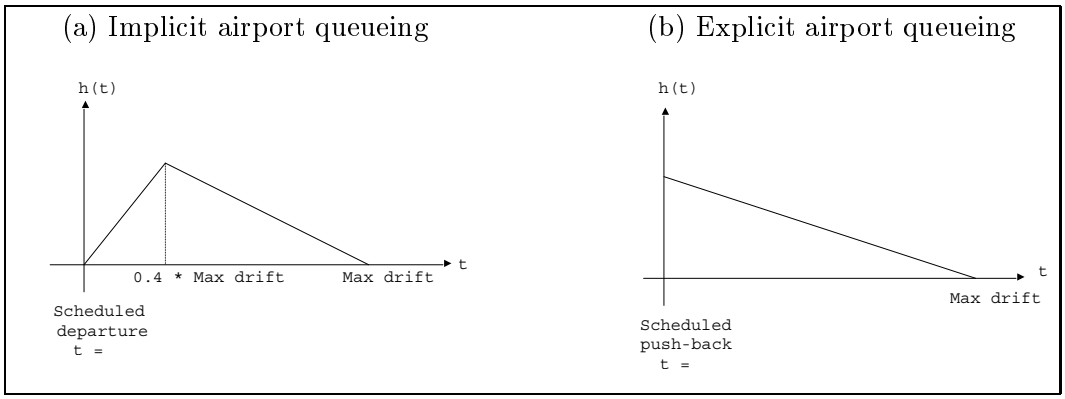


Figure 5.4: Drift probability density functions.

5.5.3 Drift Scenarios

We test the model for five drift scenarios:

1. Low constant (stationary) drift (max drift of the order of 10 min.).

2. Medium constant drift (max drift of the order of 30 min.).

3. High constant drift (max drift of the order of 60 min.).

4. Medium time-varying drift with low variance.

5. Medium time-varying drift with high variance.

5.5.4 Cancellation Scenarios

We test the model for three types of cancellations:

1. No cancellation.

2. Constant (stationary) cancellation probability (of the order of 10%). Area of the departure probability density function is normalized for each aircraft to $(1 - p)$, where p is the probability of cancellation of that aircraft.

3. Time-varying cancellation probability.

5.6 Experimental Setup- Network 1

The model and simulation are run for a duration of 500 minutes, far more than the required time for any congestion prediction application. This is to detect any consistent errors in the model that might not show up in a shorter time span. It is assumed that the system starts with no aircraft initially. This assumption does not affect the performance of the model in general, since the current position of an aircraft in the network could conceivably be modeled as

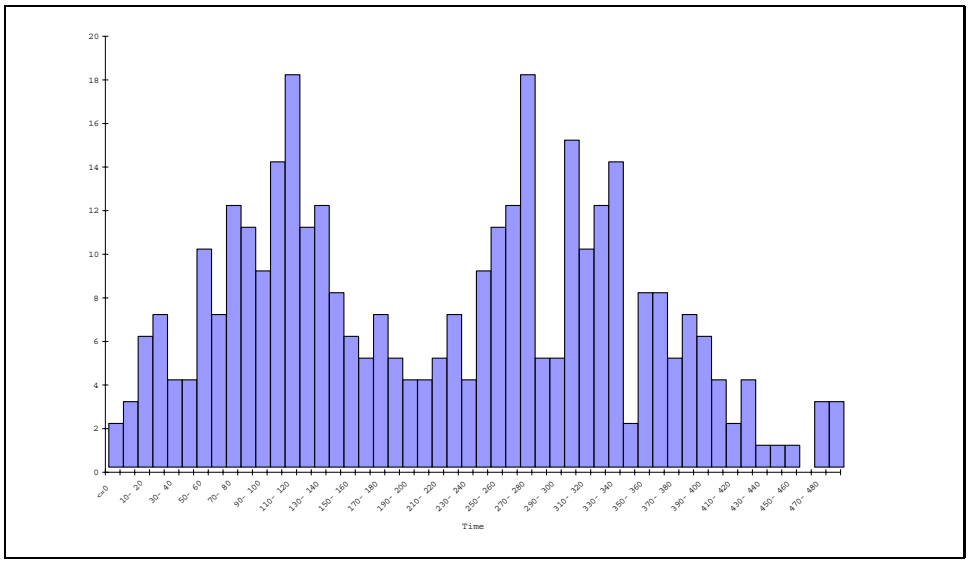


Figure 5.5: Histogram of total number of scheduled departures in network 1 over time.

a source node in the network. The length of one period was set to one minute (there are 500 minutes in each scenario set). The reason for choosing this for the period length is that the approximations used in the occupancy distributions seem to work best when the period length is approximately equal to the miles-in-trail applied. Miles-in-trail typically range from half a minute to about two minutes. The schedule of flights was generated in such a way that each airport carries approximately the same load, and there are approximately an equal number of each possible origin-destination pair. The number of departures in the network over time is shown in Figure 5.5. The two pronounced peaks were intentionally generated to resemble a typical schedule in the NAS. All computations were performed on a Sun Ultra 10 workstation on a Solaris 7 platform and all times reported are CPU times.

5.7 Experimental Results - Network 1

The standard error of the simulation (defined in Section 5.4.2) for the network (drift type 2, capacity scenario 3, drift scenario 2) for sector 1 is

$$\sigma_M = \frac{1.442}{\sqrt{200}} \frac{1}{9.56} = 1.07\%$$

where the average standard deviation over all time periods is 1.442, the mean over all time periods is 9.56 and the number of replications of the simulation is 200. We will not display all of the standard errors but note that this value is representative of the standard deviation of all other scenarios. The Monte Carlo simulation for all the scenarios for network 1 were run for 200 replications.

The required output of the model is the number of aircraft in each sector in each time interval. Two metrics of comparison are obtained for each sector - the modified RCI metric, and a simple squared deviation metric, each normalized by the area of the sector count curve. An example of such a curve is shown in Figure 5.6. Note that the curve is not continuous, but consists of a discrete number of aircraft for each time interval. The RCI and squared deviation values for the two sectors are also shown.

We ran the model, the fluid approximation, and the simulation for each of the scenarios described in the previous section (66 in all). We believe that it is not meaningful to try and draw conclusions as to the trend behavior of the model based on the relatively few observations available. For example, if we observe that the RCI metric increases as capacity increases, all other factors being constant, we might be led to draw erroneous conclusions due to the limited scope of results. We believe that the right way to draw conclusions from such a study would be test the statistical significance of the difference in

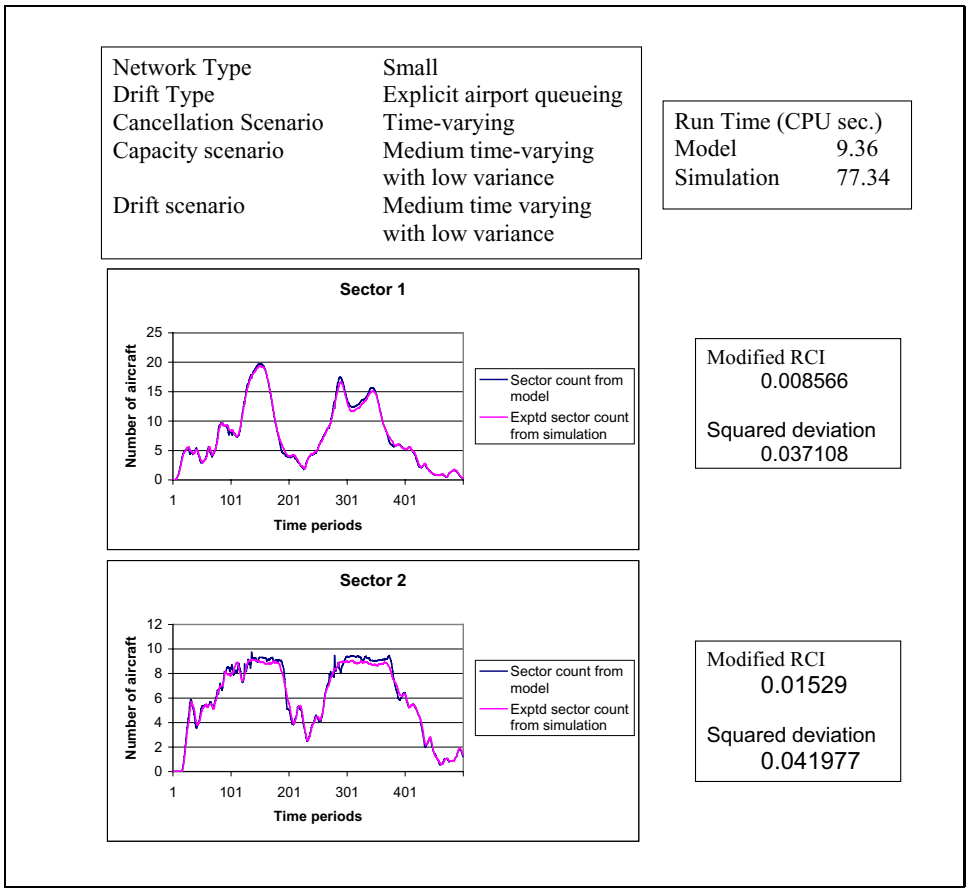


Figure 5.6: Sample output for one set of scenarios.

one type of scenario from another. We employ a technique known as a pairwise comparison hypothesis test. Details regarding the math behind the tests and the applications and interpretations are available in the volume by Albright, Winston, and Zappe [2]. For example, we can propose the null hypothesis that the RCI metric increases as the capacity increases, and, based on the p-value obtained from a statistical test, accept or reject that hypothesis. This approach is much more robust because we do not have to hold all other parameters constant while comparing the performance relative to a certain parameter.

We test some basic properties to ensure that the behavior of the models

is as expected. In particular, we observe that in the unconstrained case (no capacity restrictions at the waypoint or at the airport), the RCI metric value is extremely low. This confirms the fact that the model is exact in the unconstrained case. We can never expect the RCI value to be exactly equal to zero due to the noise in the simulation, and the fact that the period length in the experiments is finitely large (equal to one minute).

Hypothesis 1

The first question we are interested in answering is whether the model using occupancy distributions gives us a significant advantage over the fluid approximation for all the extra computation performed. We compare the RCI metric for Sector 1 for the fluid approximation and our model to determine which is superior. Our null hypothesis is that the fluid approximation is superior. It is sufficient to determine the quality of the algorithm from one sector alone as, in this case, an error in one sector implies an error in the other due to conservation of aircraft mass. The results of this hypothesis test are in Table 5.1.

Table 5.1: Results for hypothesis test H1: The RCI value for the model is greater than the RCI value for a fluid approximation.

Hypothesis	**p-value**	**Result**
RCI (Model) $\geq$ RCI (Fluid)	0.000	Reject at 10 % significance

Result 1. The model outperforms the fluid approximation in predicting expected sector counts.

Hypothesis 2

We next check to make sure that the model has a lower running time than the simulation. We hypothesize that the model does not have a lower running time than the simulation. The results of this hypothesis test are in Table 5.2.

Table 5.2: Results for hypothesis test H2: The runtime for the model is greater than the runtime for the simulation.

Hypothesis	**p-value**	**Result**
Runtime(Model) $\geq$ Runtime(Sim)	0.000	Reject at 10 % significance

Result 2. The model outperforms the simulation in terms of runtime.

Hypotheses 3, 4, 5

We would like to know the effect that the cancellations have on model performance. The hypothesis is that the cancellation scenario affects the performance of the algorithms. The results of these hypothesis tests are in Table 5.3.

Table 5.3: Results for hypothesis test H3, H4, H5 : The RCI metric obtained for different cancellation scenarios is different.

Hypothesis	**p-value**	**Result**
RCI(Canc1) $\neq$ RCI(Canc2)	0.563	Cannot reject at 5 % significance
RCI(Canc2) $\neq$ RCI(Canc3)	0.758	Cannot reject at 10 % significance
RCI(Canc1) $\neq$ RCI(Canc3)	0.409	Cannot reject at 10 % significance

Results 3,4,5. Cancellations do not affect the performance of the model.

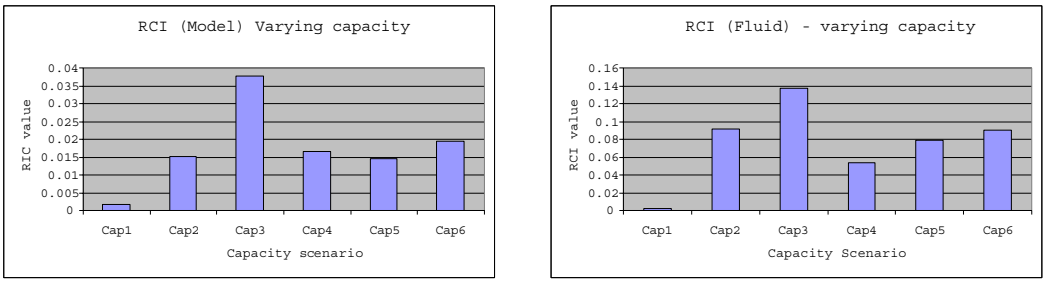


Figure 5.7: RCI values for different capacity scenarios (refer Section 5.5.1).

Hypothesis 6

We attempt to study the behavior of the system as the capacity becomes more constrained at the waypoint. We study this only for the drift type where there is no queueing at the runway. This is done to isolate the behavior of exactly one queue. The hypothesis is that the capacity scenario affects the performance of the algorithms. Due to the small sample size, it is not possible to conduct a meaningful statistical hypothesis test. Hence, we simply compare the means of the RCI metric at different levels of capacity. Although the results are intuitive and appealing, we caution against drawing strong conclusions from such a small set of results. Plots of the RCI value against different capacity scenarios for the model and the fluid approximation are shown in Figure 5.7.

Result 6. The performance of the model appears to deteriorate as the complexity of queueing (probability of interactions between aircraft) increases. However, the model easily outperforms the fluid approximation on the same instances.

An interesting observation is that the curves are not monotonous. Beyond a certain level of traffic, the performance of the models appears to improve (in figure 5.7, the model does better in Capacity scenario 4 than in

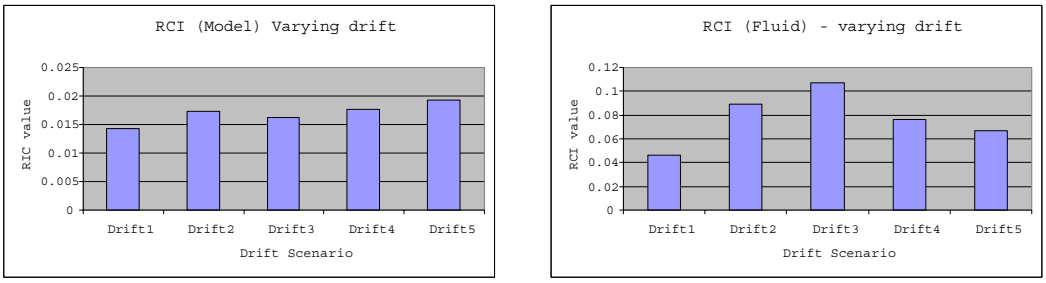


Figure 5.8: RCI values for different drift scenarios (refer Section 5.5.3).

Scenario 3, even though 4 is more tightly constrained). A possible cause for this is that as the traffic increases, the models benefit from heavy traffic limit conditions, as predicted by several authors (Newell [27]). As the level of queueing increases, the model tends to the fluid approximation.

Hypothesis 7

Next, we study the behavior of the model as the drift increases. We do this for the case where the drift does not implicitly assume airport queueing. The hypothesis is that the drift affects the performance of the model and the fluid approximation. Plots of the RCI value against different drift scenarios for the model and the fluid approximation are shown in Figure 5.8.

Result 7. The model does not seem as sensitive to the drift as the fluid approximation. Also, the model easily outperforms the fluid approximation on the same instances.

Again, we caution against drawing strong conclusions from the results of a few instances.

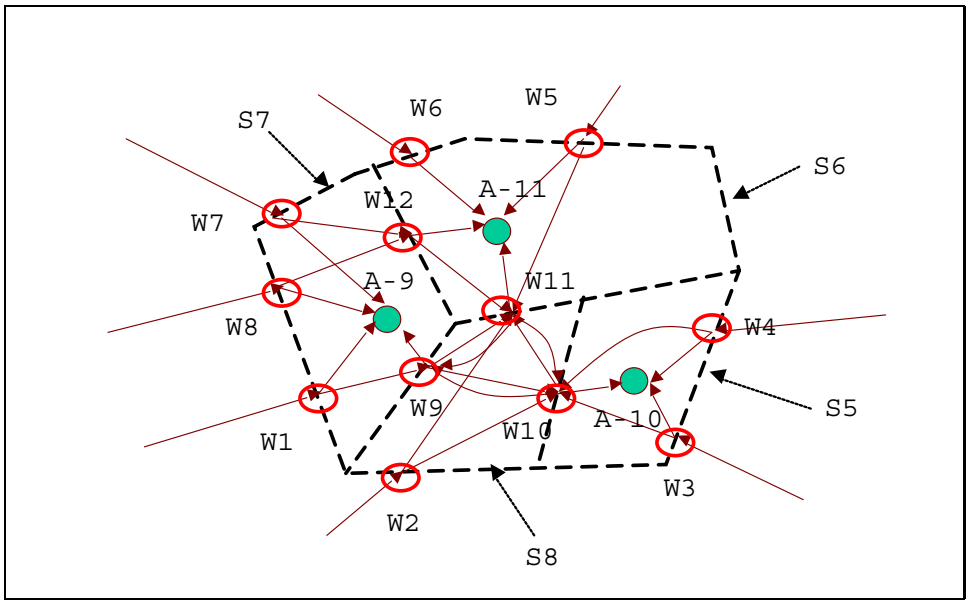


Figure 5.9: Test network 2 [W - waypoint, A - airport, S - sector].

5.8 Experimental Design - Network 2

In this section, we study the performance of the model on a more complex network with more aircraft movements. This network is illustrated in Figure 5.9. The larger test network tries to mimic actual flows in a region of airspace. In general, airspace congestion initially occurs at/near an airport/set of airports, and propagates to other regions of the airspace. An example if this is the New York region in the Northeast United States, where congestion is caused by a high density of large airports. In our example, we have three arrival airports, fed by a number of arrival streams. Queueing can occur at a number of waypoints, and an aircraft typically passes through 3-4 waypoints once it enters the region of interest in the network. We expect the model to perform better on this network than the smaller test network, as we expect errors to be compensated by multi-directional flows in the network.

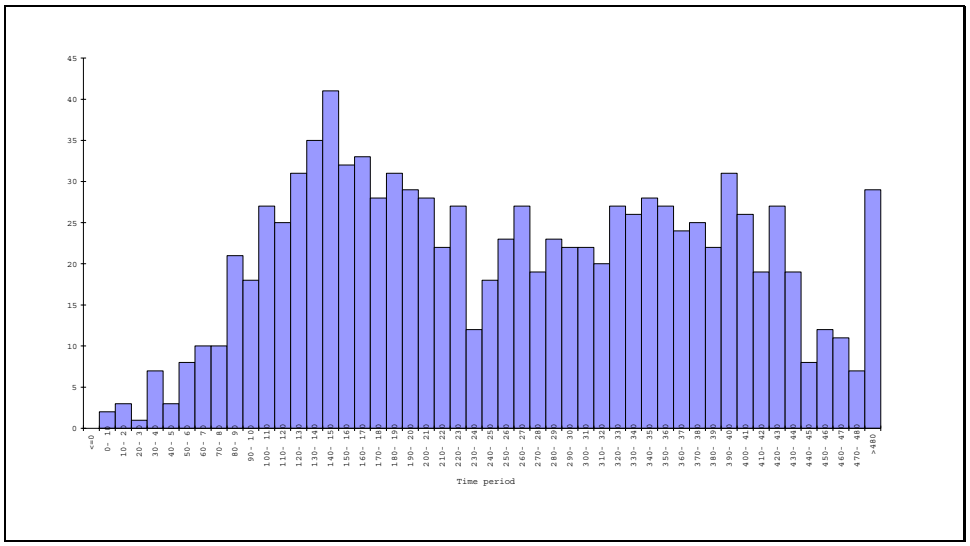


Figure 5.10: Histogram of total number of scheduled departures in network 2 over time.

We believe that performing an analysis similar to that of the smaller network would not be useful as it would be hard to isolate the contribution of each queue to the overall error and hence would not be of much use in terms of learning anything new about the behavior of the model. Instead, we demonstrate the working of the algorithm on a contrived set of scenarios, which closely resemble the actual environment in which the model is designed to be used. We analyze the network under conditions of time-varying drift (as in all real-world scenarios). Since the flows into the network are not directly from airports, drift type 2 is used (as illustrated in Figure 5.4(a)). The number of departures is similar to the previous set of scenarios, with two significant peaks in the departure schedule. The number of departures over time summed over all airports is shown in Figure 5.10. Four scenarios are investigated:

1. A "base" mode where the network is completely unconstrained to ensure

the validity of the model.

2. A scenario where the network has only nominal constraints. These are the minimum miles-in-trail required for airspace safety. These capacities do not vary with time - all airports have a capacity of 60 runway operations/hour, and a miles-in-trail of 20 seconds is applied at each waypoint over the entire time horizon of 500 minutes. None of the flights are cancelled.

3. A scenario where the airports have time-varying capacity such that the capacity of the airport is significantly degraded for a small period of time and recovers (as often occurs in the case of thunderstorms affecting an airport). This causes significant queueing at the airports, and high sector counts in sectors containing these airports. The capacity scenarios for the three arrival airports A-9, A-10, and A-11 (refer Figure:5.9) are shown in Figure 5.11. The miles-in-trail at all waypoints is set equal to the nominal MIT (20 seconds). We assume that this reduction in capacity causes flights to have a time-varying probability of being cancelled that is related to the capacity structure. The probability of cancellation of a flight over time is shown in Figure 5.12.

4. In the final scenario, some time-varying control (miles-in-trail) is applied at the waypoints to mitigate the congestion in the previous scenario. This is shown in Figure 5.13.

The fluid approximation is also applied to each of the above four scenarios. These scenarios are developed in an attempt to demonstrate the actual framework within which our model could be used.

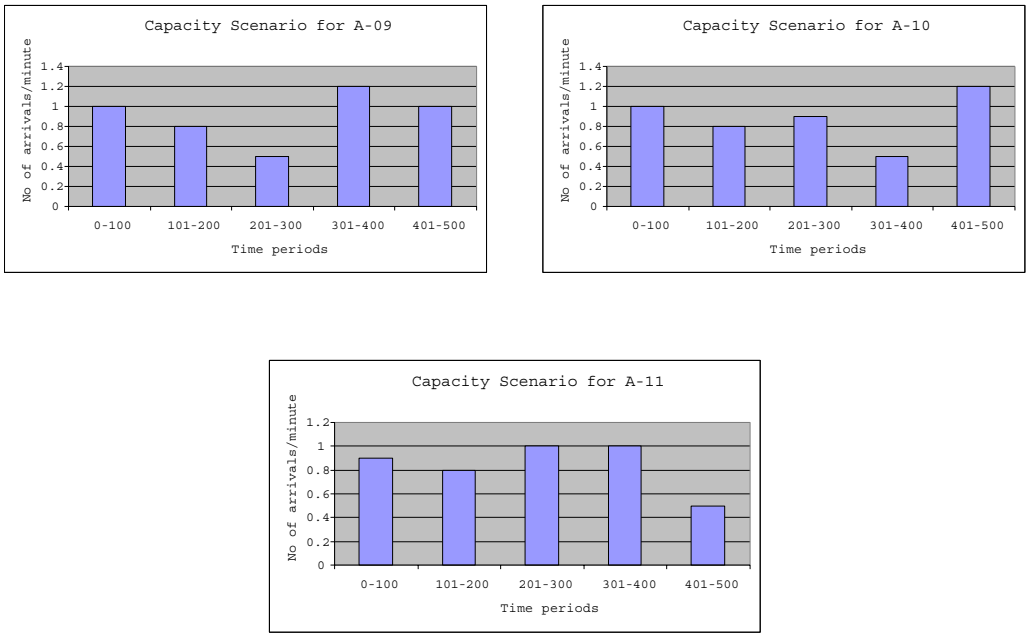


Figure 5.11: Reduced capacity scenarios for airports A-9, A-10, and A-11.

5.9 Experimental Results - Network 2

To provide an indication of the simulation accuracy, Table 5.4 gives the standard error of the simulation (defined in Section 5.4.2) for the network (scenario 2) for the four sectors. For example, the standard error of sector S5 was computed as

$$\sigma_M = \frac{1.750}{\sqrt{500}} \frac{1}{8.036} = 0.97\%$$

where the average standard deviation over all time periods is 1.442, the mean over all time periods is 9.56 and the number of replications of the simulation is 500. The Monte Carlo simulation for all the scenarios for network 2 were run for 500 replications.

In this section we present the computational results for each of the scenarios described in the previous section. The output consists of plots of

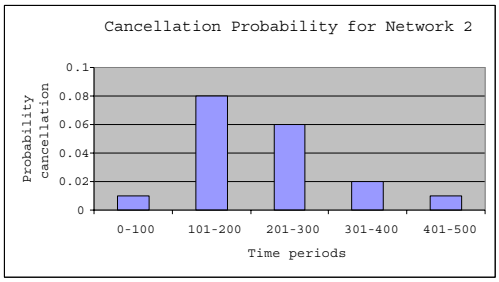


Figure 5.12: Cancellation probabilities for network 2 caused by reduced capacity.

Table 5.4: Standard error of mean for the simulation of network 2 for scenario 2 (nominal constraints).

Sector	Std. Error (%)
S5	0.97
S6	0.81
S7	0.80
S8	1.16

expected sector counts over time, and the corresponding RCI values for these sectors. In addition, we randomly sample a set of 7 flights that enter the network at different times in the time horizon, and compute the expected estimated time of arrival (ETA) for each of these aircraft using the simulation and the model. The same is done for the fluid approximation.

5.9.1 Scenario 1 (Unconstrained)

The results for the model and the fluid approximation are presented in Figure 5.14 and Figure 5.15 respectively. As expected, both the model and the fluid approximation are almost exact as a consequence of Theorem 3.6.1.

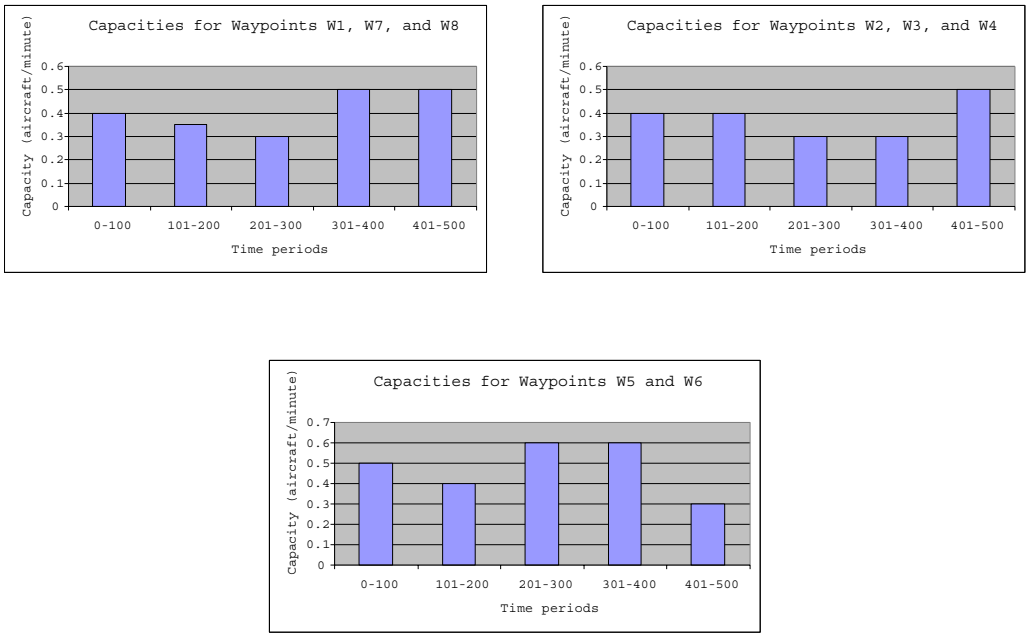


Figure 5.13: Reduced capacity at waypoints W1 through W8 in response to congestion.

5.9.2 Scenario 2 (Nominal Constraints)

This scenario is to test the level of utilization in the airspace under "normal" conditions. The results of this scenario using the model and the fluid approximation are presented in Figure 5.16 and 5.17 respectively.

It is clear from the results that the fluid approximation significantly underestimates queueing, while the model estimates queueing delays accurately. This is reflected in the low RCI values as well as the fact that the ETA from the fluid approximation is consistently earlier than the ETA from the simulation. Another observation is that the sector counts are at or slightly above the capacities recommended (refer Table 1.1). This implies that any degradation in capacity of the airports will cause the sector counts to violate sector capacity constraints.

5.9.3 Scenario 3 (Capacity Reduction at Airports)

This scenario tries to mimic a realistic capacity scenario where each of the airports' capacity is degraded in a time-varying fashion. Such a capacity degradation is commonly encountered when an airport is affected by a thunderstorm. The results from the model and the fluid approximation for this scenario are presented in Figures 5.18 and 5.19.

It is clear from these results that the model outperforms the fluid approximation (the RCI values for sectors S5, S6, and S7 are significantly lower for the model than the fluid approximation). It also appears that the model tends to *overestimate* congestion in cases of very high queueing. The sector counts obtained from the model and the fluid approximation clearly indicate that the sector capacity constraints are very likely to be violated in sectors S5, S6, and S7.

5.9.4 Scenario 4 (Controls Applied to Mitigate Congestion)

In the previous scenario, the sector capacities of three sectors are likely to be exceeded significantly. This calls for some intervention from the traffic flow managers/ controllers/ airlines to mitigate this congestion. In this scenario, we assume that the only reaction to this congestion comes from the flow managers who set some time-varying miles-in-trail at the waypoints leading into the network (waypoints W1 - W8) in order to decrease the rate of flow of aircraft into the congested sectors. The traffic flow manager attempts to mitigate congestion by applying some miles-in-trail at the incoming fixes (W1 to W8).

The reduced capacities of the waypoints are shown in Figure 5.13. The results from the model and the fluid approximation for this scenario are presented in Figures 5.20 and 5.21.

Based on the RCI values, it is clear that the model outperforms the fluid approximation in this scenario. Another observation (also seen in Scenario 3) is that the model sometimes tends to overestimate queueing, while the fluid usually tends to underestimate queueing.

We also observe that the flow management initiatives (miles-in-trail applied at incoming fixes (W1 to W8) have still not solved the congestion problem. Sector S5 remains congested, implying that greater miles-in-trail needs to be applied at W3 and W4. Congestion in sector S6 has been mitigated in the earlier half of the time horizon, though stricter control probably needs to be applied to control congestion in the second half of the time horizon. Congestion in sector S7 has been reduced considerably.

The objective of the above scenario was not to eliminate congestion, but to demonstrate the use of the model as a decision support tool that would enable traffic flow managers analyze the effects of different traffic flow initiatives.

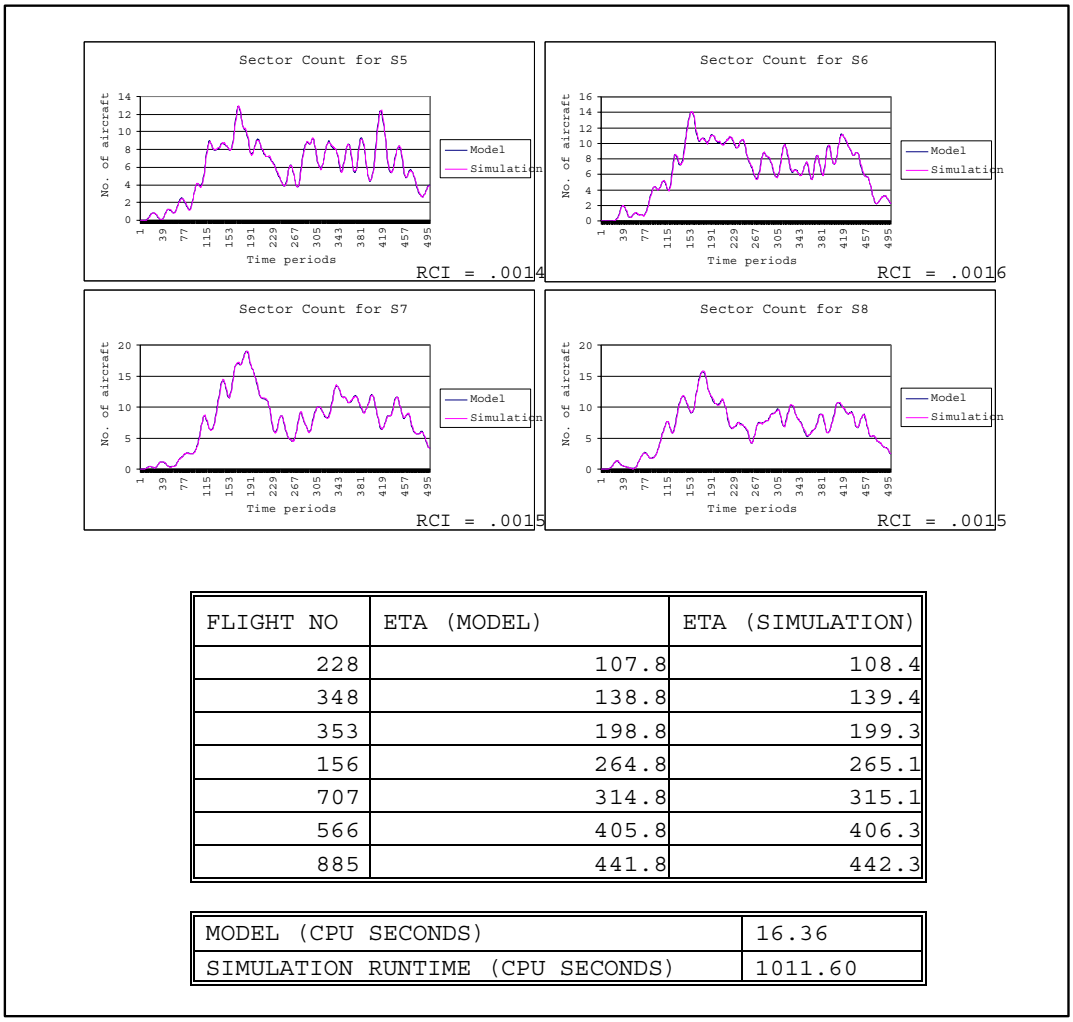
5.10 A Note on the Confidence Interval on the Model Output

In order for any prediction of a variable to be complete in a probabilistic setting, it should not only have a mean, but an associated variance as well, so that the spread of this distribution is known. In the examples above, we were

concerned primarily with predicting the *expected* number of aircraft in a sector at any given time. If the model were run on some arbitrary day, the actual realization of the sector counts could be different from the expected value (this would correspond to one run of the Monte Carlo simulation). Hence, we would like to place some confidence interval on our prediction so that it is more complete in describing the sector count. The model cannot, as such, predict the variance associated with the sector count at some time. A possible approach to the problem is to estimate the variance approximately using the knowledge of the number of packets, and their masses, which are available from the model. We recall that our interpretation of a packet mass is the probability of the existence of a packet at a given point in time and space. Thus, we essentially have a number of packets (entities) which have a binary state based on a probability (either in the sector or not in the sector). If all the masses of the packets were identical, this would correspond to a binomial distribution of n trials with a success probability of m, where n is the number of packets in the sector at a given time, and m is the mass of each packet. In our model, however, the masses of each packet are different, and hence this is not a classical binomial distribution. However, if the number of packets were large enough for the variance in the packet mass to be inconsequential, then we could obtain a binomial distribution for the sector count with parameters n, the number of packets, and $\overline{m}$, the mean mass of all the packets. It is also known that as the number of packets increases, the binomial distribution can be approximated by a normal distribution. Hence, given a large number of packets with similar mass, we can obtain the mean and variance of a corresponding binomial/ normal curve, which would give a confidence interval

on the prediction. In this thesis, we focus on the mean of the sector counts, and not the variance.

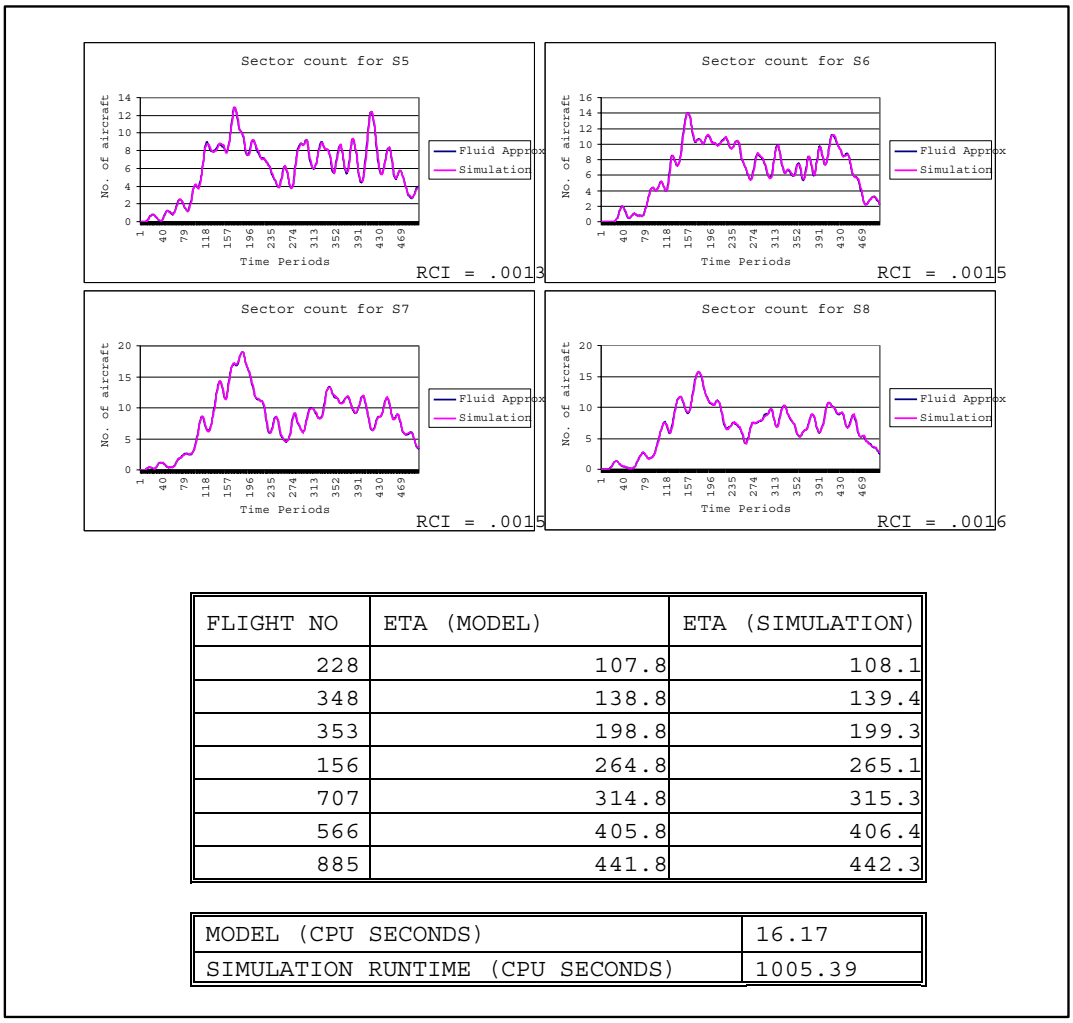
We observe for our computational experiments for network 2 that the standard deviation in the sector counts over all runs of the Monte Carlo simulation are of the order of 1 to 2 aircraft. The expected sector count varies from 10 to 40 aircraft. Hence, a congestion prediction based on the expected value alone would still be useful since one or two aircraft would not significantly change a congestion prediction, especially if averaged over a 15 minute interval (as in the case of Monitor Alert - refer Section 1.2.4).



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	107.8	108.4
348	138.8	139.4
353	198.8	199.3
156	264.8	265.1
707	314.8	315.1
566	405.8	406.3
885	441.8	442.3

MODEL (CPU SECONDS)	16.36
SIMULATION RUNTIME (CPU SECONDS)	1011.60

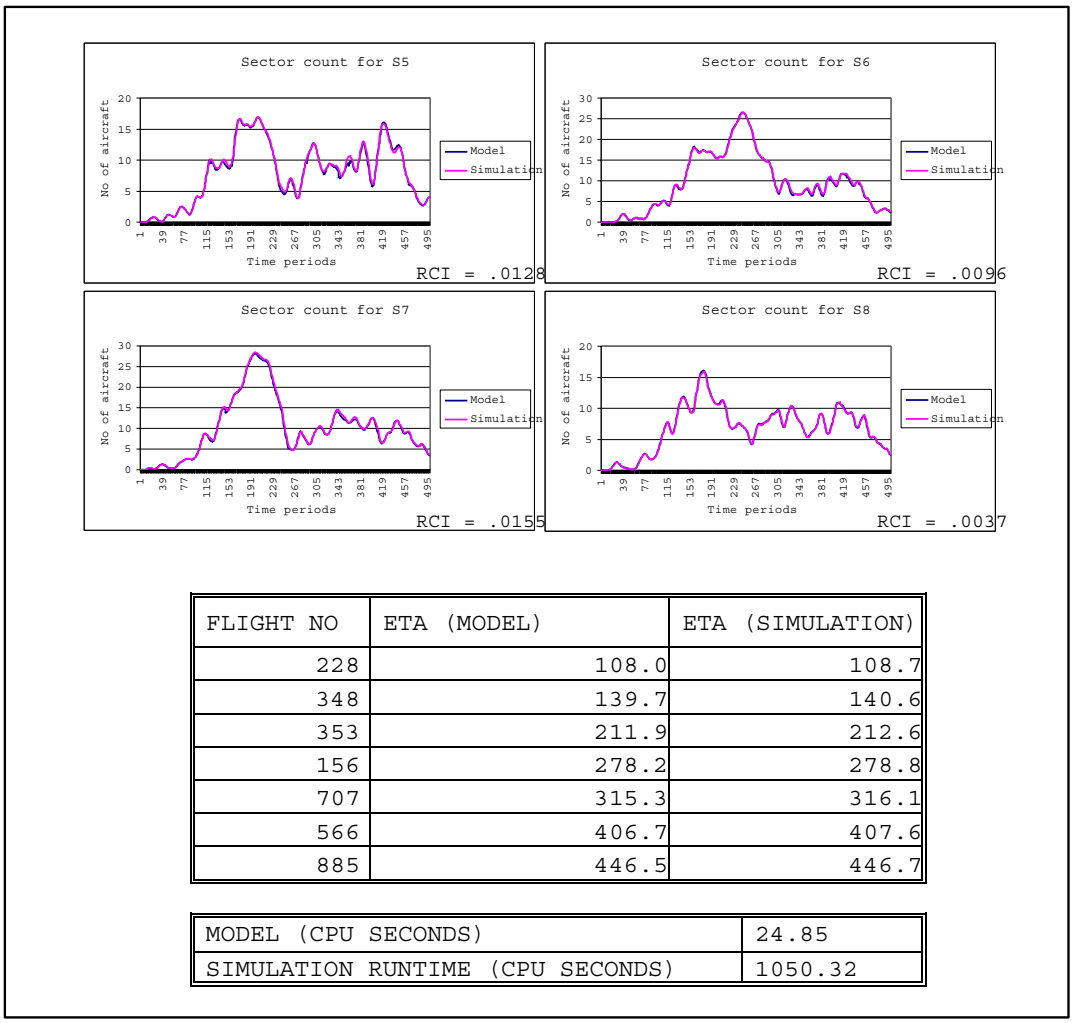
Figure 5.14: Results of scenario 1 (unconstrained) for network 2 using the model.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	107.8	108.1
348	138.8	139.4
353	198.8	199.3
156	264.8	265.1
707	314.8	315.3
566	405.8	406.4
885	441.8	442.3

MODEL (CPU SECONDS)	16.17
SIMULATION RUNTIME (CPU SECONDS)	1005.39

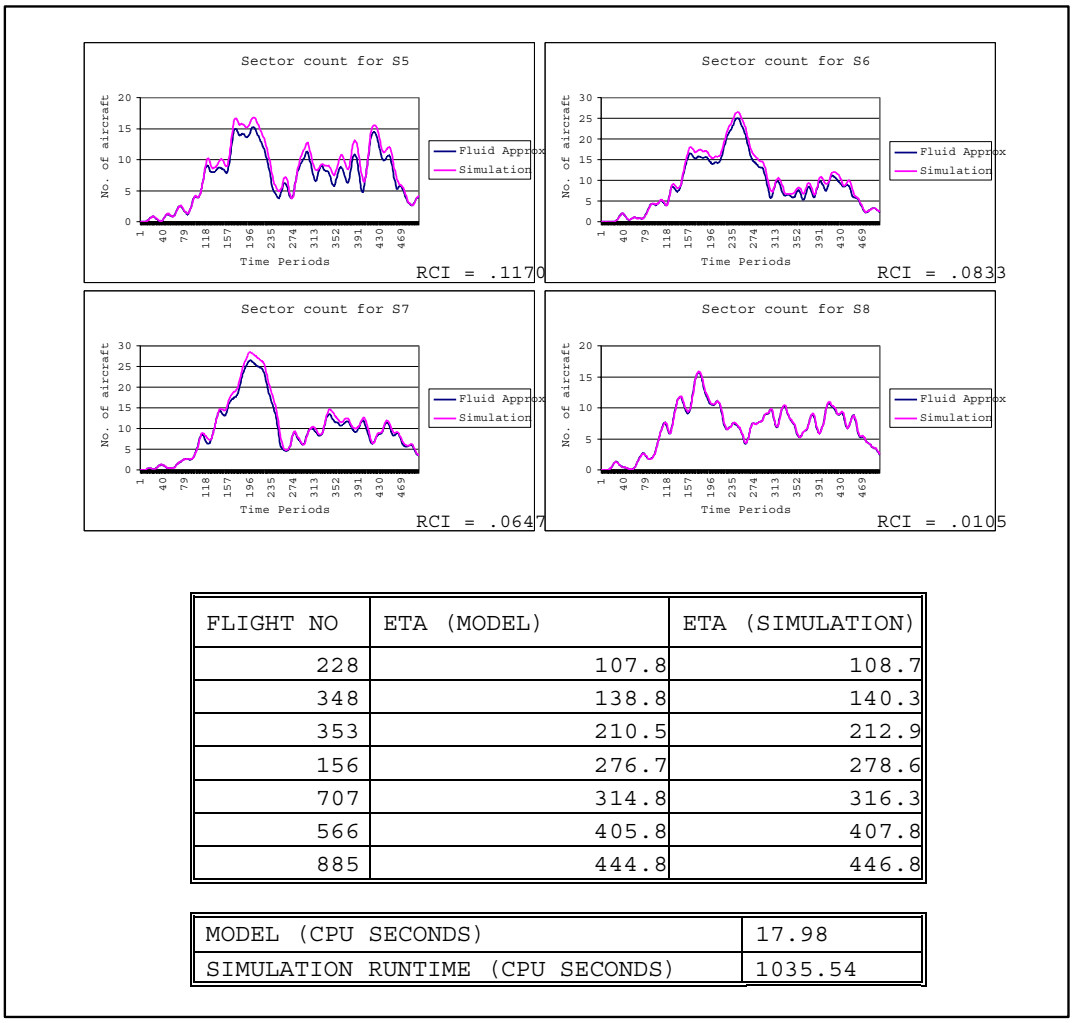
Figure 5.15: Results of scenario 1 (unconstrained) for network 2 using the fluid approximation.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	108.0	108.7
348	139.7	140.6
353	211.9	212.6
156	278.2	278.8
707	315.3	316.1
566	406.7	407.6
885	446.5	446.7

MODEL (CPU SECONDS)	24.85
SIMULATION RUNTIME (CPU SECONDS)	1050.32

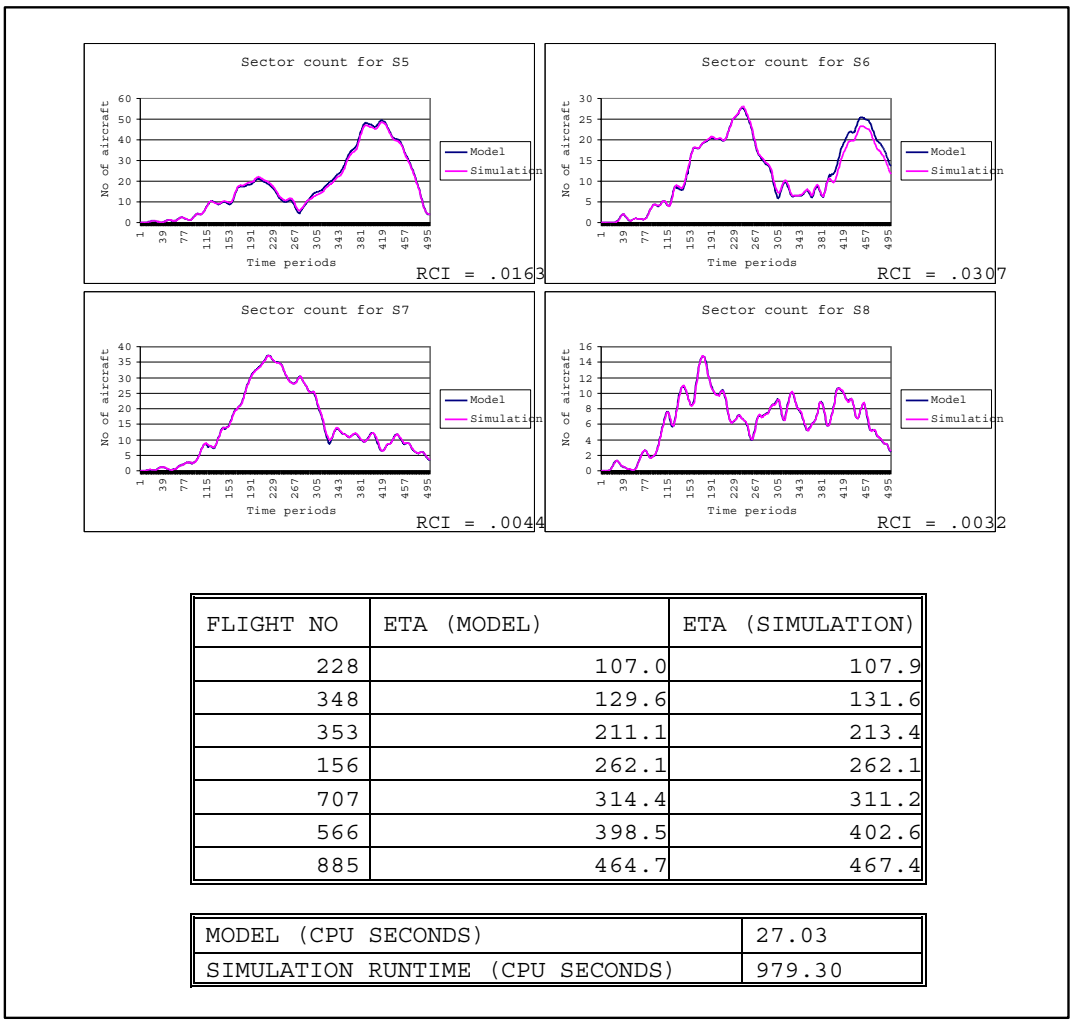
Figure 5.16: Results of scenario 2 (nominal constraints) for network 2 using the model.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	107.8	108.7
348	138.8	140.3
353	210.5	212.9
156	276.7	278.6
707	314.8	316.3
566	405.8	407.8
885	444.8	446.8

MODEL (CPU SECONDS)	17.98
SIMULATION RUNTIME (CPU SECONDS)	1035.54

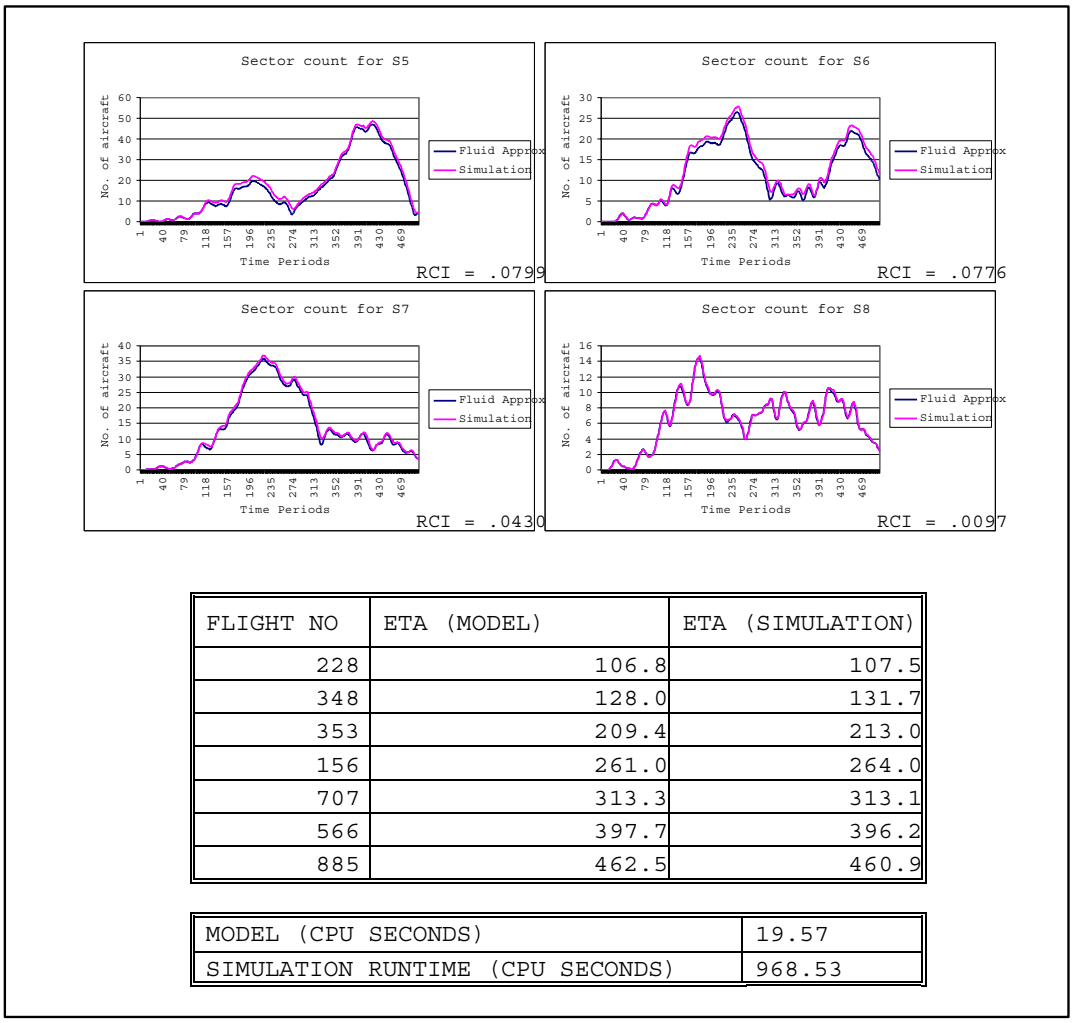
Figure 5.17: Results of scenario 2 (nominal constraints) for network 2 using the fluid approximation.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	107.0	107.9
348	129.6	131.6
353	211.1	213.4
156	262.1	262.1
707	314.4	311.2
566	398.5	402.6
885	464.7	467.4

MODEL (CPU SECONDS)	27.03
SIMULATION RUNTIME (CPU SECONDS)	979.30

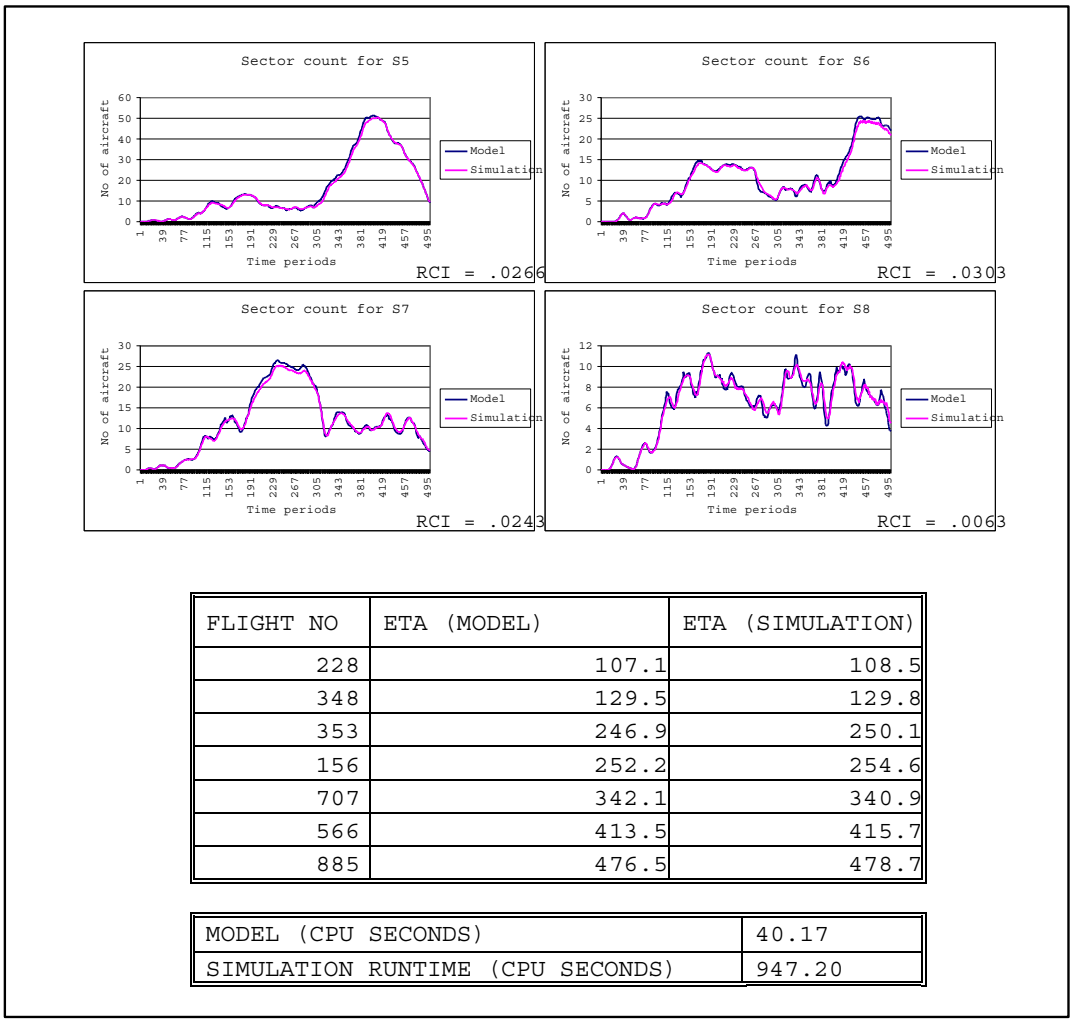
Figure 5.18: Results of scenario 3 (reduced arrival capacity) for network 2 using the model.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	106.8	107.5
348	128.0	131.7
353	209.4	213.0
156	261.0	264.0
707	313.3	313.1
566	397.7	396.2
885	462.5	460.9

MODEL (CPU SECONDS)	19.57
SIMULATION RUNTIME (CPU SECONDS)	968.53

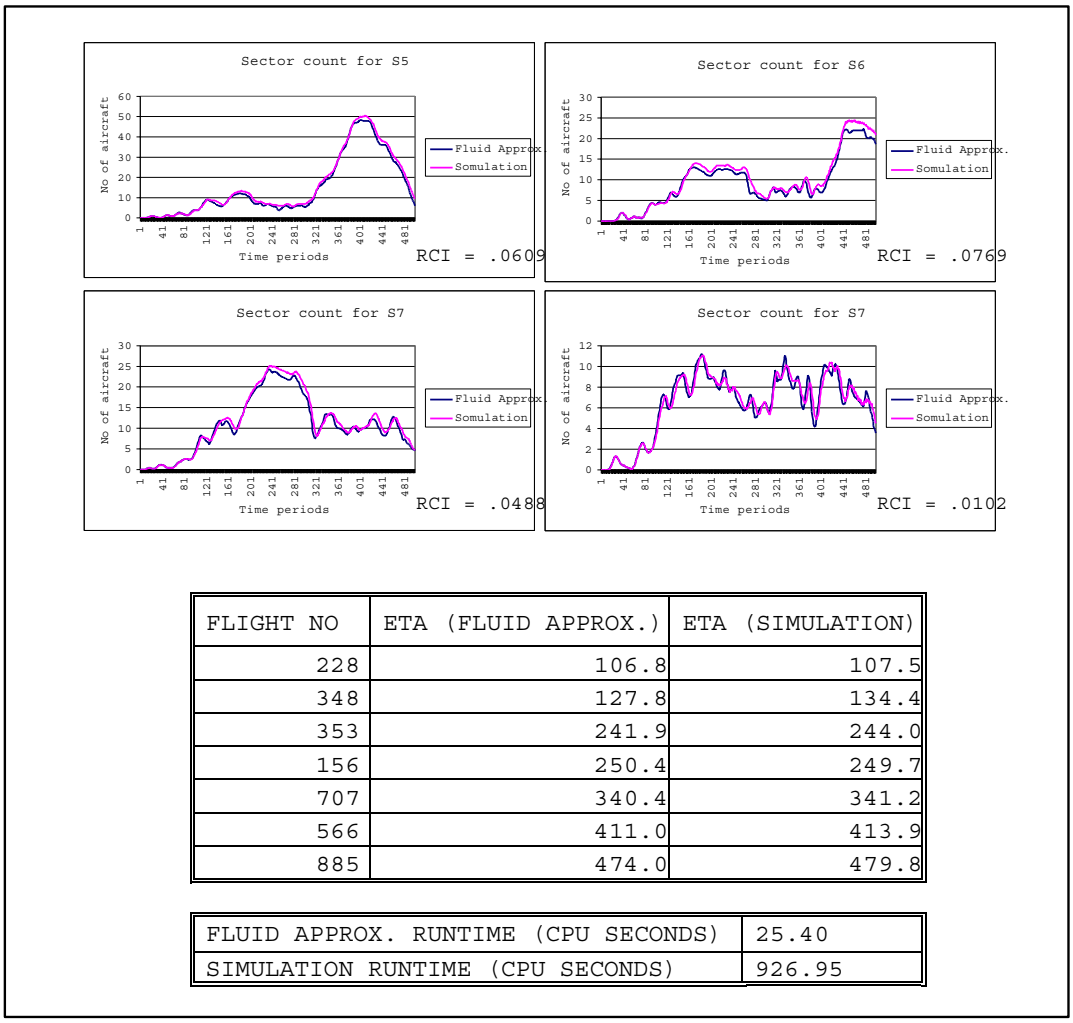
Figure 5.19: Results of scenario 3 (reduced arrival capacity) for network 2 using the fluid approximation.



FLIGHT NO	ETA (MODEL)	ETA (SIMULATION)
228	107.1	108.5
348	129.5	129.8
353	246.9	250.1
156	252.2	254.6
707	342.1	340.9
566	413.5	415.7
885	476.5	478.7

MODEL (CPU SECONDS)	40.17
SIMULATION RUNTIME (CPU SECONDS)	947.20

Figure 5.20: Results of scenario 4 (controls applied in response to congestion) for network 2 using the model.



FLIGHT NO	ETA (FLUID APPROX.)	ETA (SIMULATION)
228	106.8	107.5
348	127.8	134.4
353	241.9	244.0
156	250.4	249.7
707	340.4	341.2
566	411.0	413.9
885	474.0	479.8

FLUID APPROX. RUNTIME (CPU SECONDS)	25.40
SIMULATION RUNTIME (CPU SECONDS)	926.95

Figure 5.21: Results of scenario 4 (controls applied in response to congestion) for network 2 using the fluid approximation.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We have presented a model for the analysis of queueing delays in highly dynamic networks with schedule-based stochastic arrivals and time-varying capacities. Two types of schedule uncertainty were addressed:

1. Uncertainty in the time of birth of an entity in the system.

2. Uncertainty in the fact that the entity is born (cancellations).

The model was tested on a large number of scenarios on a small network with varying problem parameters. The model was also applied to a larger representative network to observe the performance of the model under "real-world" conditions. Conditions under which the model is exact were investigated, and demonstrated empirically.

A fluid approximation was developed, where the problem was solved as a simple network flow model. The performance of the fluid approximation was analyzed over the same test instances.

A sub-problem of the original problem (generating server occupancy distributions) was also analyzed, and exact results presented for some special cases. An approximation technique was developed to estimate occupancy distributions, and successfully incorporated into the model.

6.2 Application

The model was successfully implemented and applied to a sample problem of estimating queueing delays in the airspace. The performance of the model was compared to a simulation of the same network. The example in Section 5.9 demonstrated the environment in which the model is expected to operate. The model could be used either as a congestion prediction tool, or as a decision support tool that enables decision makers (flow managers, airlines) evaluate the impact of proposed changes in schedules, flightpaths, and miles-in-trail on the level of congestion and delays.

6.3 Other Applications

The model is not restricted to the airspace alone, and could conceivably be applied on a wide variety of similar networks. We believe that the model could be applied to at least one other type of problem that occurs in manufacturing systems. Manufacturing usually follows a planned schedule of production, subject to uncertainty in adherence to this schedule. Although service times usually are not very dynamic, the model could be used to predict queueing delays in such an environment. Readers interested in applications of queueing theory in manufacturing are referred to [28].

Many of the approximations used in the model were developed specifically keeping typical airspace traffic in mind. If the model were designed to be used in a different setting, two major parts of the model will have to be re-calibrated:

- Occupancy distributions - The regression equations presented in Section 4.3.3 will have to be re-estimated.
- The value of γ in Equation 3.9 will have to be adjusted for the "typical" level of traffic.

6.4 Future Work

Occupancy distributions were estimated using linear regression. We believe that a more accurate and robust approach to this problem would be to develop a neural network model that generates occupancy probability distribution parameters based on model inputs.

We envision that the model will ultimately be available to traffic flow managers as well as airlines. Since each player tries to maximize efficiency with incomplete information regarding the schedules and proposed traffic flow initiatives, the model presents extensive gaming opportunities to all players. Hence, the model would eventually have to be coupled with a game-theoretic model of the network, which analyzes the effect of incomplete information on the decisions taken by each player in the network.

A fairly recent development in the characterization of airspace congestion has been the concept of dynamic density [24]. The approach advocates that a simple threshold number, such as the Monitor Alert Parameter, does not

sufficiently capture the complexity of traffic within a sector to accurately identify congestion. The concept of dynamic density tries to incorporate the complexity of traffic into a congestion metric, thus giving more accurate congestion predictions in terms of controller workload. We believe that our model could be used in such a setting, as our model keeps track of individual aircraft, and its characteristics. It is thus possible to obtain from the model an aggregate number of aircraft belonging to each type of flow in a sector and other aircraft characteristics such as the aircraft type that contribute to the dynamic density metric. Once a clear definition of dynamic density emerges, some research would be necessary to map the model output onto a dynamic density metric.

Bibliography

[1] M. Abramowitz and A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables.* Dover, New York, 1972.

[2] S. Albright, W. Winston, and C. Zappe. *Data Analysis and Decision Making with Microsoft Excel.* Duxbury Press, Pacific Grove, CA, 2002.

[3] M. Ball, G. Gosling, and A. Odoni. *Workshop summary: Airline and National Strategies for Dealing with Airport and Airspace Congestion.* College Park, Maryland, 2001.

[4] M. Ball, T. Vossen, and R. Hoffman. Analysis of demand uncertainty in ground delay programs. In *Proceedings of 4th USA/Europe Air Traffic Management R & D seminar*, 2001.

[5] F. Baskett, K. Chandy, R. Muntz, and F. Palacios. Open, closed and mixed networks of queues with different classes of customers. *J.ACM*, 22:248–260, 1975.

[6] E. Beaton, J. Brennan, J. DeArmon, J. Formosa, K. Levin, S. Miller, and C. Wanke. Predicting congestion in the northeast U.S.: A search for indicators. *3rd USA/Europe Air Traffic Management R & D Seminar*, June 2000.

[7] G. Bell. Theoretical studies into traffic congestion, part 2. Technical Report 3, British Ministry of Civil Aviation, Operations research Section, ORS/MCA, June 1948.

[8] W. Beyer. *CRC Standard Mathematical Tables.* CRC Press, Boca Raton, FL, 28 edition, 1987.

[9] B. Chandran. Airspace congestion prediction tool: Algorithms and code documentation. Available on request, 2002.

[10] FAA. National airspace system documentation. Internet, retrieved 17 July, 2002. http://204.108.10.116/INDEX.HTML.

[11] FAA. Order 7210.3, facility operation and administration. Internet, retrieved 22 May, 2002. http://www.faa.gov/atpubs/FAC/Ch17/s1706.html#17-6-1.

[12] Marcos Escobar Fernandez de la Vega. *Approximate Solutions for Multi-Server Queueing Systems with Erlangian Service Times and an Application to Air Traffic Management.* PhD thesis, Massachusetts Institute of Technology, 1998.

[13] H. Galliher and R. Wheeler. Nonstationary queueing probabilities for landing congestion of aircraft. *Operations Research*, 6(2):264–275, 1958.

[14] E. Gelenbe and G. Pujolle. *Introduction to Queueing Networks.* John Wiley & Sons, Chichester, England, 2 edition, 1987.

[15] L. Green, P. Kolesar, and A. Svoronos. Some effects of nonstationarity on multiserver markovian queueing systems. *Operations Research*, 39(2):502–511, 1991.

[16] R. Hoffman. *Integer Programming Models for Ground-Holding in Air Traffic Flow Management*. PhD thesis, University of Maryland, College Park, 1997.

[17] R. Hoffman and M. Ball. The rate control index for traffic flow management. *IEEE Transactions on Intelligent Transportation Systems*, 2:55–62, 2001.

[18] H. Idris, J. Clarke, R. Bhuva, and L. Kang. Queueing model for taxi-out time estimation. submitted to ATC quarterly, 2001.

[19] J. Jackson. Jobshop like queueing systems. *Management Science*, 10:131–142, 1963.

[20] M. Jambunathan. Some properties of beta and gamma distributions. *Annals of Math. Stat.*, 25:401–405, 1954.

[21] L. Kleinrock. *Queueing Systems: Theory*. John Wiley & Sons, New York, USA, 1975.

[22] P. Kostiuk, D. Lee, and D. Long. Closed loop forecasting of air traffic demand and delay. In *Third USA/Europe Air Traffic Management R&D Seminar*, Napoli, June 2000.

[23] I. Krysicki. On some new properties of the beta distribution. *Stat. Prob. Let.*, 42:131–137, 1999.

[24] I. Laudeman, S. Shelden, R. Branstrom, and C. Brasil. Dynamic density: An air traffic management metric. Technical Report NASA/TM–1998-112226, NASA Ames, 1998.

[25] K. Malone and A. Odoni. The approximate network delays model. Working paper, 2001.

[26] W. Massey and W. Whitt. Networks of inifinite-server queues with non-stationary poisson input. *Queueing Systems*, 12:183–250, 1993.

[27] G. Newell. *Applications of Queueing Theory.* Monographs on Statistics and Applied Probability. Chapman and Hall, New York, USA, 2 edition, 1982.

[28] H. Papadopoulos, C. Heavey, and J. Browne. *Queueing theory in manufacturing systems analysis and design.* Chapman & Hall, UK, 1993.

[29] T. Pearcey. Delays in landing of air traffic. *Journal of the Royal Aeronautical Society*, 52:799–812, 1948.

[30] M. Peterson, D. Bertsimas, and A. Odoni. Decomposition algorithms for analyzing transient phenomena in multiclass queueing networks in air transportation. *Operations Research*, 43(9):995–1011, 1995.

[31] M. Pike and I. Hill. Algorithm 291: logarithm of the gamma function. *Communications of the ACM*, 9:984, 1966.

[32] T. Saaty. *Elements of Queueing Theory With Applications.* Dover Publications, 1961.

[33] L. Schaefer and D. Millner. Flight delay propagation analysis with the detailed policy assessment tool. In *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference*, volume 11. ASME Press, 2001.

[34] W. Voss and J. Hoffman. Analytical identification of airport and airspace capacity constraints. *3rd USA/Europe Air Traffic Management R & D Seminar*, June 2000.

[35] P. Zehna. *Probability distributions and statistics.* Ally & Bacon, Inc., Boston, 1970.