ESTIMATING MIXTURES OF EXPONENTIAL DISTRIBUTIONS USING MAXIMUM LIKELIHOOD AND THE EM ALGORITHM TO IMPROVE SIMULATION OF TELECOMMUNICATIONS NETWORKS

by

SEAN ROBERT BAIRD

B.Com., The University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE (BUSINESS ADMINISTRATION)

in

THE FACULTY OF GRADUATE STUDIES

(Department of Commerce and Business Administration; Operations and Logistics Division)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 2002

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMMERCE & BUSINESS.

The University of British Columbia
Vancouver, Canada

Date DEC. 20 / 2002

DE-6 (2/88)

Abstract

This thesis explores the topic of mixture-distributions as they relate to modeling call demand on a telecommunications network. Modeling call duration demand in particular proves difficult for a number of reasons. Historically, this has been modeled using a simple exponential distribution with a single parameter. This work extends that modeling technique to using multi-component exponential distributions. Development of these models is shown to be possible using non-linear programming as well as an application of the EM algorithm. These independent approaches yield remarkably similar results. Also relevant are the treatment of statistical significance testing for large data set samples, since these notoriously pose difficulty by magnifying statistical significance. This problem is treated through a more robust comparison of data to the theoretical distribution using a bootstrapping technique of sampling against the large data set. Finally, the results of the demand modeling are also validated using a more intuitive comparison to the simulation model output.

Table of Contents

List of Tables

List of Figures

Acknowledgement

First, I would like to thank my wife, Alyson for supporting me throughout my pursuit of this Master's degree. Without her sincere encouragement and practical advice, I could not have completed this.

I would like to acknowledge and thank Professor Martin Puterman for being my advisor on this project and research. Without your seed ideas and assistance in modeling, I could not have completed this thesis. Your consistent support, and subtle challenge to improve made this work much stronger. I would also like to thank Jonathan Berkowitz for his sincere advice and consideration with respect to my work. I appreciate your efforts.

Sincere thanks to Stephen Jones and Paul Hiom for their assistance with all things COE throughout my life as a graduate student. Conversations with the two of you were instrumental in shaping my career at UBC. Also, thanks to my fellow project members, Thaddeus Sim and Kelly Chung for tireless efforts and brainstorming contributions.

Finally, thank you to my friends at the COE – Lauren Gray, Mehmet Begen, Pia Bustos, Omar Ladak, Kevin Tang, and Kyle Biswanger – to name a few. Without your encouragements, diatribes, and everything in between, this experience could not have been as rich.

1. Introduction

1.1. Telecommunications

Research in modern telecommunications technology is fast-paced and productive when it comes to designing systems for communications. However, much of this research is never fully implemented due to the massive capital costs of making changes to existing infrastructures. Since the invention of the telephone, the telecommunications industry has progressed surely and steadily towards providing service to many customers. However, along the way, technological changes to make the systems more efficient have been limited to very specific areas. Existing telecommunications networks around the world still use what is termed *Circuit-Switching* technology to connect calls between parties. This terminology often conjures up images of switchboard operators manually interconnecting individual circuits to complete telephone calls. In fact, this same concept is still the case today, albeit with certain advances.

As an example of these advances, switching speed and capacity has progressed through three phases of technology. Originally, manual interconnection of circuits was performed by operators; this was gradually replaced by large mechanically operated switching devices, and more recently, today's modern digital switches which can connect circuits electronically. In the same way, transmission has improved from allowing a single call on a twisted-pair line to allowing up to 96 simultaneous calls to share a circuit through a process known as multiplexing, by which the frequencies of each call are modulated before transmission, so as not to interfere with each other, and demodulated after transmission.

The applications of these technological advances, while allowing for greater transmission through existing equipment, conceal a deeper issue of concern with the circuit-switched system. In order to develop this, I will begin by explaining exactly how circuit-switching works. Circuit Switching is the process by which an incoming call is assigned a series of interconnections in order to create a constant connection – circuit – between the origin and destination parties. This process is in use today by most modern telecommunications networks.

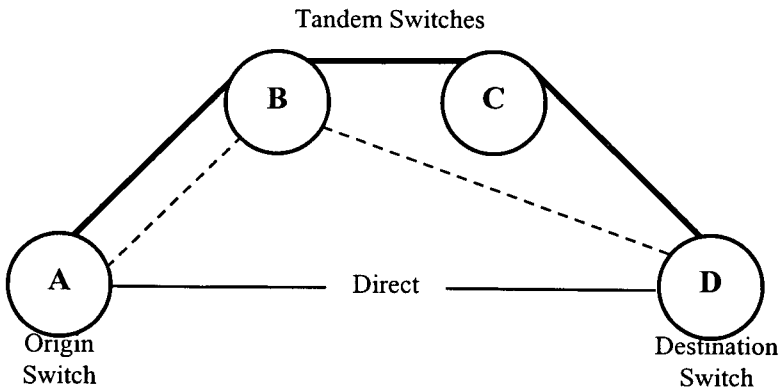


Figure 1-1: An Example of a Circuit Switched Network

Figure 1-1 provides a graphical representation of the most common switching protocol used in Circuit-Switching, termed *Alternate Hierarchical Routing*. Under this protocol, should a call originating at switch A be destined for switch D (A & D are termed an *OD pair*), it can follow one of three possible routes[1]. First, the routing system will attempt to

[1] Note that routing may actually be accomplished through software distributed amongst the various switches, or through a dynamically aware central networking system.

route the call directly to switch B (the *Primary Route*). Should this first route be blocked[2], the routing system will sequentially try the route from A-B-D, and finally A-B-C-D. These secondary routes are termed *tandem routes*, since they utilize a second (or tandem) circuit. If the call cannot be connected upon the final route (termed the *Alternate Final)*, it is blocked and the originating customer receives either a fast busy-signal, or an "all circuits are busy" message. It should be noted that this example and all subsequent studies herein deal only with the non-toll local calling backbone, resulting in the following assumptions and repercussions:

1. No price is charged to individual calls, either for revenue, or priority purposes.
2. Connections between telephones (remotes), and the backbone are not examined
3. Switches are assumed to have instantaneously infinite capacity

In circuit-switching, the capacity of any arc between switches is a function of the number of circuits installed. Although it is theoretically possible to compress 96 calls onto a single connection through the above-mentioned process of modulation, most modern telecommunications companies (TELUS included) modulate only 24 calls for reasons of security and quality. As a result, the total number of calls which may be carried directly between switch A and B is the number of lines installed multiplied by 24, this is generally referred to as the OD pair's trunk group size. The concealed issue of concern mentioned above is this requirement that each call utilize an entire circuit between switches. Notice that this problem is compounded if the call goes through one of the tandem switches,

---

[2] Blockage occurs when there are no available circuits between a pair of switches.

causing the call to utilize two or more entire circuits for its duration. However, many telephone calls do not require constant transmission of data. People conducting a conversation will pause, fax machines will take time to process data, individuals using dial-up internet connections may spend time reading a web-page without sending or receiving data. Yet all this time, while the circuit is not being utilized, it is unavailable for transmitting other demand on the same OD pair.

The most remarkable changes to telecommunications are coming in the form of a new transmission technology. This new system called packet switching will allow more efficient use of capital equipment by transmitting data only when necessary, and breaking it up into small packets each of which can follow independent routes from origin to destination. In this manner, no constant connection is required between the switches. Provided there is a path available when a packet needs to be transmitted, there is no degradation of service.

1.2. Research Context

Many telephone companies, TELUS included, are pursuing a move towards using packet switching for regular voice traffic. However, while the technology is meant to greatly enhance the capacity of transmission medium, it is also incompatible with current switches. As a result, TELUS is faced with a situation where although demand for voice telephone calls is still steadily increasing, any investments they make to maintain the system will likely become obsolete once the migration to packet switching gets underway.

My research is part of a larger project that was conducted through the Center for Operations Excellence (COE) at the University of British Columbia. Throughout this project, we work with the traffic engineering department at TELUS in order to increase the efficiencies of the existing circuit switched network, so as to mitigate the need for increased investment. Figure 1-2 shows an overview of the COE/TELUS project. The focus of this thesis will be on the process referred to as *Distribution Modeling*, and its impact primarily upon the network simulation model and its efficiency. For information specifically on the Optimization and Simulation processes, please see Sim (2001), Kabanuk (2000), Smith (2000), and Braun (1999).

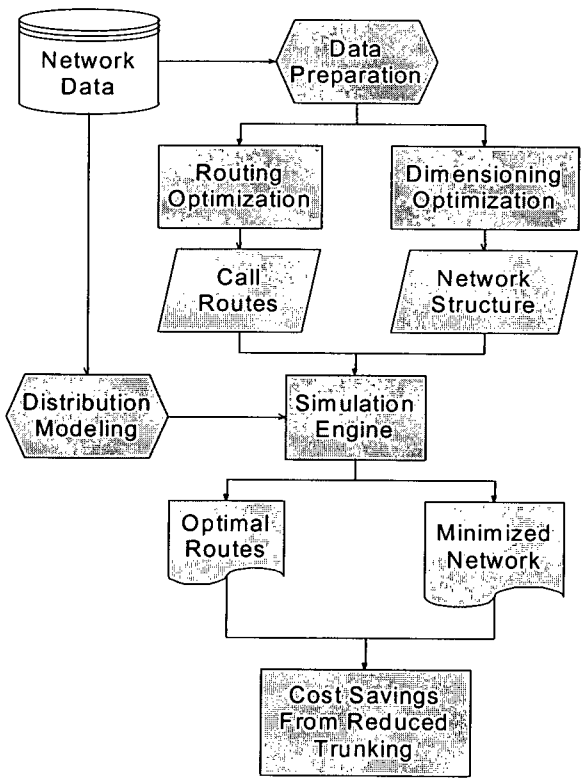


Figure 1-2: An overview of the COE / TELUS research project

1.3. Simulation and Distribution Modeling

One drawback of making efficiency improvement suggestions in the telecommunications-networking field is that each suggestion has the potential to have an enormous impact upon service delivery. Any change to the network infrastructure or routing rules will undoubtedly impact the probability of blockage between switches, which is a performance metric monitored by regulatory authorities overseeing telecommunications in many countries. While recommendations would generally be made to reduce this blockage level, the Canadian Radio-television and Telecommunications Commission (CRTC) imposed a regulatory standard of *p.01* (Less than 1% of calls blocked during the busiest hour of the busiest day) means that it is neither economically, nor managerially feasible to make physical changes to the network or routing rules in order to measure the operational impacts of our optimization applications. This gave rise to the idea of using a test-bed simulation. In his thesis, Braun (1999) describes the creation of a Monte-Carlo Circuit Switched Network Simulator which replicates the behavior of the network under varying demand conditions. Many modifications have subsequently been made to the underlying simulation engine, including recently the addition of a user-friendly front-end in order to facilitate use of the simulator as an impact analysis tool by the Traffic Engineering Department at TELUS. However, until recently, the simulator ran using empirical distributions for both the call arrival rate and call holding times. These empirical distributions were used simply because they were simple to program into the simulation, easy to generate, replicated reality, and preliminary attempts at fitting theoretical distributions proved unfruitful.

Conventional wisdom in the field of simulation and probability suggests that it would be preferable to generate call demand from a theoretical distribution for two primary reasons:

1. *The theoretical distribution should be more representative of population characteristics* – Empirical distributions are discrete, since they are generated directly from samples, while the true parameters underlying call demand are continuous. Therefore, values not represented in the sample will not appear when generating data with empirical distributions.
2. *Theoretical distributions are easier to work with than massive volumes of data* – Simulation using an empirical distribution requires a sequential search function, which is generally less efficient than a closed-form CDF calculation.

It follows then, that these theoretical distributions should be capable of providing a more accurate view of the behavior of our observed random variables. In addition, generating call demand data from theoretical distributions has a profound impact on the speed of the simulation procedures, since it is much faster to calculate values from the closed-form cumulative density function than to look-up empirical values in a data array. However, Bratley, Fox and Schrage (1983) caution that using a theoretical distribution should be based on *a priori* justification of these distributions. Therefore, in the case of telephone call distributions, we should consider classic queuing theory, such as Poisson arrival processes, and exponential holding times.

1.4. Potential Data Models

The classic application of Poisson arrival processes and exponential holding times has worked well for telecommunications modeling in the past. In fact, Section 5.1 will show that this theory still holds well for applications of the arrival process. However, Section 5.2 will show that simple exponential holding time patterns do not necessarily reflect the data collected and analyzed from TELUS. The symptoms of this problem, are the so-

called "fat-tailed distributions" resulting from the introduction of longer overall telephone calls. However, this problem can be decomposed even further. Longer call holding times are the result of specific usage patterns, i.e., the rise in data-calls associated with modem-based internet access. In theory, when separated from each other, different call-types (personal calls, business calls, data calls, credit-card verifications, etc) should each exhibit exponential holding time distributions. Unfortunately, the type of call is not recorded, and thus this information is not available for modeling. Therefore, we must use some method of estimating how many constituent call types there are, what proportion of each type make up the whole, and what their individual component densities are. Together, this information represents a *mixture density*, so called because it is simply a proportionally-weighted mixture of the individual component densities. Section 3 goes into more detail about how these densities are decomposed and estimated. Section 4.2 describes the application data sets for which this technique is used.

2. Evaluation Criteria

Before attempting to fit a theoretical distribution to any data, it is important to determine what criteria will be used to evaluate its use. I have approached this process in three ways, where the distinctions are subtle yet profound. First, we can validate the fit of the theoretical distribution by ensuring that the empirical data matches the theoretical distribution within some tolerance level. Alternatively, we can validate the use of a theoretical distribution by simulating the random variables multiple times, and ensuring that the resulting data are consistent with the original sample data used to parameterize the distributions. Finally, we can implement the theoretical distributions in the network

simulator, and check that the simulation results are consistent with those created by the empirical distribution, or in reality.

The difference between these approaches is that the first uses a statistical method which relies on tests with notoriously low power (Bratley, Fox and Schrage, 1983), while the second method uses a more intuitive, but less rigorous notion of simply examining the results of data generation and comparing them to existing data. Both of these methods can and should be applied to both in-sample and out-of-sample data.

2.1. Statistical Goodness of Fit Tests

Conover (1999) writes the distribution goodness of fit hypotheses as follows:

$$H_0 : F(x) = F^*(x) \qquad \text{for all } x \text{ from } -\infty \text{ to } +\infty$$
$$H_a : F(x) \neq F^*(x) \qquad \text{for at least one value of } x$$

Where F^* is the "true" data-generating distribution from which we have taken a sample $x_1, x_2, ..., x_n$. The goodness of fit tests can be thought of as testing whether some hypothesized distribution (F) produced the given set of sample data. The two most widely used statistical goodness of fit tests are the Chi-Squared test and the Kolmogorov-Smirnov test.

Although the Chi-Squared test's low power results in only a small likelihood of rejecting an inappropriate theoretical distribution, the Chi-Squared statistic is simple to calculate, and will result in rejecting the hypotheses that certain theoretical distributions produced data from some of our switch pairs should we utilize an inappropriate model. The test is performed by binning the empirical data and comparing the frequency counts in each of

these bins to the levels expected given the theoretical distribution. Thus, for n distinct data points:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where :

O_i = The observed proportion of observations in bin i

E_i = The expected proportion of observations in bin i under the theoretical distribution

This equation produces a test statistic which if the null-hypothesis is true $\left(F(x) = F^*(x)\right)$ has a chi-squared distribution with $n-k-1$ degrees of freedom, where k is the number of parameters estimated.

The Kolmogorov-Smirnov test utilizes the Empirical Distribution Function (EDF). An EDF is generated by ordering the observed data, such that $x_1 \leq x_2 \ldots \leq x_i \leq x_{i+1} \leq \ldots \leq x_n$. The EDF, $F(x)$, is the fraction of observations which are less than or equal to x, for every possible x. These EDF goodness of fit tests then test whether some supposed distribution $F^*(x)$ is significantly different from the observed EDF, $F(x)$.

The Kolmogorov-Smirnov Test concerns the entire empirical distribution. In order to encompass the whole distribution, this test looks at the maximum vertical deviation between the observed EDF, and the proposed distribution function $F^*(x)$. The test statistic T, is calculated as the highest absolute deviation between the two functions:

$$T = \sup\left|F^*(x) - F(x)\right|$$

Thus, the null hypothesis $\left(F(x) = F^*(x)\right)$ is rejected at significance level α if T exceeds the $1-\alpha$ value of the Kolmogorov-Smirnov distribution, which must be looked up in a table. Similarly, one can specify a confidence band for a given EDF as opposed to actually testing its fit.

Connover and others suggest that the use of either Chi-Squared or Kolmogorov-Smirnov tests can be problematic when faced with a particularly large sample sizes. These problems come from the fact that when a goodness of fit test can examine a multitude of observations, any deviation from the expected distribution can be amplified, thereby showing statistical significance, but not necessarily practical significance. As a result, two methods of further investigation are possible. First, the distribution may be tested repeatedly against smaller samples of the available data (a bootstrapping technique). Second, a simulation method may be used to test the practical impact of utilizing a hypothesized distribution on many other output variables.

2.2. Validation through Simulation

The second method used to validate the choice of distribution is less mathematically rigorous, but more relevant to the problem at hand. The purpose of fitting these distributions is to improve the simulation of call demand. As accurate as any of the previously mentioned tests may be, they do not present any metric that would be recognizable or useful to the average user of traffic engineering data. As an alternative method of validation, the previously mentioned Circuit-Switch Network Simulator examines the impact on the entire system of using the hypothesized distribution.

In his thesis, Braun (1999) indicates that the simulation uses flat files, storing the empirical distributions in order to generate the call demand. A simple test of the distribution "usefulness" could be designed as follows:

1. Estimate the parameters of the theoretical distributions (See Section 5 for examples of this in practice).
2. Modify the simulation in order to draw from these theoretical distributions
3. Compare the output of the simulation with actual recorded data about the behavior of the network (See Section 6.1 for practical application).
4. Examine the impact of the new model on decision-making (See Section 6.2 for practical application).

Now, provided that the choice of theoretical distribution parameters has a significant effect upon the observations of the simulation, one can test somewhat more holistically whether the proposed distribution functions are adequate. This test may be more useful than a simple mathematical test. In addition, it also provides a framework for comparing the current functionality of the simulation with the proposed changes.

In addition, this method of testing encompasses the interaction that is inherent in the results of applying different types of call demand data. Where the EDF tests can only examine the impact of using a particular call demand distribution, testing through simulation examines the effect of both call duration and call arrival distributions across all possible switch pairs. In effect, this methodology creates a simultaneous test which is both easier to compute and more useful than the aforementioned mathematical tests.

The fourth element of the test, which examines the impact of changes on actual decision-making, falls naturally from the previously described process. Since the simulation engine is being used to quantitatively support decision-making within a telecommunications-

networking environment, a logical question is what impact the proposed changes would have upon the decision-making process and product. These effects will be estimated by examining the particular decisions currently being supported, and how their quantitative output is affected by the changes to the simulation model.

3. Maximum Likelihood Estimation

In order to provide distributions of call demand data for the Circuit-Switched Network Simulator, some method must be used to estimate the distribution parameters. Both standard (one-component) probability densities and mixture (multi-component) probability densities have to be estimated. When estimating standard probability distributions, this can be interpreted as determining the parameters of the underlying distribution. For example, if $x_1, x_2, ..., x_N$ are observations of a random variable with probability density $f(x, \theta_1, ..., \theta_m)$, having m parameters, then the process will be concerned with estimating $\theta_1, ..., \theta_m$. However, in the case of mixture densities, where $x_1, x_2, ..., x_N$ are drawn from K random variables with probability densities, $f^1(x, \theta_1^1, ..., \theta_m^1)$, and $f^j(x, \theta_1^j, ..., \theta_m^j)$,..., $f^K(x, \theta_1^K, ..., \theta_m^K)$ respectively, the process will be concerned with estimating the underlying distribution parameters$(\theta_1^1, ..., \theta_m^1, \theta_1^2, ..., \theta_m^2, ..., \theta_1^K, ..., \theta_m^K)$ as well as the relative proportion of observations from each random variable $(\pi^1, ..., \pi^K)$.

In their survey article, Redner and Walker (1984) show that there are many methods for generating these estimates, including maximum likelihood, the method of moments, and minimum chi-squared. However, use of these other methods was generally confined to

analytical work on mixture densities in the absence of fast and easy computing technology. Even before this technology was available, maximum likelihood estimation was considered to be the method of choice and was limited only by computational obstacles to its use. Therefore, other methods will not be considered.

Generation of maximum likelihood estimates is a non-trivial process involving maximization of a non-linear function in multiple dimensions. In the data samples used for the TELUS project, the maximum likelihood estimation required non-linear maximization in three or four variables, depending upon the number of component densities selected.

Due to the inherent problems associated with non-linear maximization, no method for generating these estimates is infallible. Therefore, I have used two independent methods in order to generate estimates, non-linear programming with the Newton-Raphson algorithm, and the Expectation-Maximization (EM) algorithm. Both of these methods are well established in scientific literature as successfully providing valid maximum likelihood estimates for many applications, including mixture density estimation.

3.1. Non-linear programming

Maximum likelihood estimation of mixture densities is generally presented in the form of non-linear maximization. For example, if $x_1, x_2, ..., x_N$ are observations of a random variable with probability density $f(x, \theta_1, ..., \theta_m)$, having m parameters, then the likelihood function ℓ is defined as:

$$\ell(\theta_1, ..., \theta_m) = \prod_{i=1}^{N} f(x_i, \theta_1, ..., \theta_m)$$

The likelihood function, $\mathcal{L}$, then represents the joint density of the individual observations for any given level of the distribution parameters. The maximum likelihood estimate is the distribution parameter values, which maximize $\mathcal{L}$

$$(\hat{\theta}_1,...,\hat{\theta}_m) = \arg\max \{ \mathcal{L} \}$$

However, since these same values of the estimators also maximize the logarithm of $\mathcal{L}$, a simple logarithmic transformation is generally made in order to avoid the product term in the objective function.

While this method works particularly well for most standard distributions, the introduction of mixture densities complicates the problem, primarily in the form of introduced constraints. For example, consider a set of observations, $x_1, x_2, ..., x_N$, drawn from K distributions, with probability densities, $f^1(x, \theta_1^1, ..., \theta_m^1)$, and $f^j(x, \theta_1^j, ..., \theta_m^j), ..., f^K(x, \theta_1^K, ..., \theta_m^K)$ respectively. If the observations are unlabelled, then we are unable to determine which random variable each observation originates from. This in turn implies that we do not know the relative distribution of observations, π_j, originating from each variable. Now, the likelihood function becomes:

$$\mathcal{L}(\pi_1,...,\pi_K,\theta_1^1,...,\theta_m^1,\theta_1^K,...,\theta_m^K) = \prod_{i=1}^{N}\sum_{j=1}^{K}\pi_j \cdot f^j(x_i,\theta_1^j,...,\theta_m^j)$$

In this scenario, the likelihood function is a function of many more parameters, including the individual parameters of each probability density, and the mixing proportion parameters, π_j. The maximum likelihood estimates are:

$$(\hat{\theta}_1^1,...,\hat{\theta}_m^1,...,\hat{\theta}_1^j,...,\hat{\theta}_m^j,...,\hat{\theta}_1^K,...,\hat{\theta}_m^K,...,\hat{\pi}_1,...,\hat{\pi}_K) = \arg\max\{\ell(\pi_1,...,\pi_K,\theta_1^1,...,\theta_m^1,\theta_1^K,...,\theta_m^K)\}$$

In order to ensure that $\pi = (\pi_1,...,\pi_K)$ is a probability distribution, we add the constraint:

$$\sum_{j=1}^{K} \hat{\pi}_j = 1$$

By reformulating this constraint as an inequality, we can eliminate one of the original decision variables in the objective function. This is done by constraining the sum of the remaining distribution parameters to be less than or equal to 1, and assigning the value of the final distribution parameter so as to make the summation add to 1:

$$\sum_{j=1}^{K-1} \hat{\pi}_j \leq 1$$

The unnecessary decision variable may now be removed. As was the case with the single standard distribution, the objective function in this case may be simplified by maximizing the logarithm of the likelihood function, since this will also maximize the likelihood function. I this way, we eliminate the unnecessary product term. Putting these two functions together, we have the basis for a non-linear programming approach to solving for the maximum likelihood estimators. This formulation uses $(K \cdot m)$ variables for the independent distribution parameters, since each of the *m* parameters must be estimated for each of the *K* variables. Similarly, this formulation requires $K-1$ variables for the mixing parameters, one for each of the components, except the last which would necessarily be redundant due to the requirement that these variables sum to one. So the optimization problem becomes:

$$\max \sum_{i=1}^{N} \left\{ log \left[\sum_{j=1}^{K-1} \left(\pi_j f^j (x_i, \theta_1^j, ..., \theta_m^j) \right) + \left(1 - \sum_{j=1}^{K-1} \pi_j \right) f^K (x_i, \theta_1^K, ..., \theta_m^K) \right] \right\}$$

$$\text{subject to } \sum_{j=1}^{K-1} \pi_j \leq 1$$

Solution of this non-linear optimization problem has long been a difficult task for applied researchers. Early attempts at solving these equations resulted in applications of Newton's method to small numbers of equations in Mendenhall and Hader (1958). However, due to its computational complexity, larger scale problems were not tackled using this method until computational power increased. Once computational power was available, researchers generally used the iterative maximization technique known as the Expectation-Maximization (EM) algorithm.

My solution of this programming problem is achieved through an application of *Newton's Method;* an example is shown in Chapter 5 for the TELUS call demand data. It can be shown that solutions using this method are extremely dependent upon initial estimates for the parameters. There are two generally accepted practices for dealing with this problem

1. Generate the maximum likelihood estimates multiple times using different combinations of original estimates in order to test the solutions robustness
2. Use another method to develop estimates of the initial parameters to within an appropriate magnitude.

Both of these methods are discussed in more detail in the application example given in Chapter 5.

3.2. Expectation-maximization (EM) algorithm

The EM algorithm is often used in circumstances involving missing data; in this case, the missing data is the identity of the originating distribution for each observation. The EM algorithm arose from many different maximization algorithms concerning specific distributions and conditions. It was first published in a cohesive and generalized manner by Dempster, Laird and Rubin in 1977. In my area of concern, estimating parameters for mixtures of distributions in the exponential family, independent research conducted by Day (1969), Hasselblad (1969), and Wolfe (1970) all made contributions to the general formulation of the EM algorithm.

As discussed above, the general problem of dealing with parameter estimation for mixtures of distributions involves missing information. Although we hypothesize that data comes from more than one source, we cannot record which source accounts for which observation.

As an example, suppose we collect data on the lifetimes of some component, but suspect that we are observing data from more than one brand. Assuming we are unable to record which brand we are observing, this data is referred to as unlabeled, and is a candidate for estimation using mixed exponential densities, since the component lifetime for each brand will likely follow exponential distributions. In this example, using the lifetimes of components, we could use the EM algorithm to estimate the proportion of components falling into each category (brand), and the distribution of component lifetime within each particular brand. Assuming we chose to mix only two distributions, this would result in

three parameters to be estimated. The first is the mixing parameter (proportion), the second and third are the means of the two exponential distributions.

The EM algorithm maximizes the likelihood that the chosen mixture distribution produced a given set of data by manipulating distribution parameters. In the case of our simple example, there are three parameters. The algorithm consists of two simple steps. In the first step, the Expectation (E-step), the probabilities that any given observation belongs to a particular category – brand in our example – are estimated using the currently fitted distribution parameters. In the second step, the Maximization (M-Step), all of the data is incorporated and the distribution parameters are re-maximized using the new estimates of the probabilities from the previous E-Step.

In his work on distributions from exponential families, Hasselblad (1969) derived the E-Step and M-Step for finite mixtures. His formulation follows:

Given the initial estimates of the mixed density parameters: $p_1^0,...,p_{K-1}^0,\lambda_1^0,...,\lambda_K^0$, then, each iteration ($v$) of the algorithm has two steps:

1) E-Step (calculate mixing proportions):

$$p_j^{(v+1)} = \sum_{i=1}^{N} \frac{f_j^{(v)}(x_i)p_j^{(v)}}{g^v(x_i)} \Big/ N$$

2) M-Step (calculate new component density parameters):

$$\lambda_j^{(v+1)} = \frac{\sum_{i=1}^{N} \frac{f_j^{(v)}(x_i) x_i}{g^{(v)}(x_i)}}{\sum_{i=1}^{N} \frac{f_j^{(v)}(x_i)}{g^{(v)}(x_i)}}$$

Where:

$f_j^{(v)}(x_i) = \lambda_j e^{-\lambda_j x_i}$	Individual exponential densities with mean λ_j at iteration v
$g^{(v)}(x_i) = \sum_{j=1}^{K} p_j \lambda_j e^{-\lambda_j x_i}$	Mixed exponential density at iteration v

Hasselblad compared the performance of the EM algorithm with another manner of obtaining estimates for a mixture density, the method of moments. While it is difficult to draw conclusions from his small study, his examples routinely showed that the EM algorithm produced estimates with smaller variances than the method of moments estimates.

4. Data Samples

Telecommunications networks are generally engineered to handle peak loads. As a result, peak calling volumes are generally used to represent network demand. Instantaneous call volume can be thought of as the amount of calling traffic moving across the network at any given time. For the purposes of most modeling, call volumes are treated as a single dimension, volume-based variable, representing a constant flow, or average demand level. However, in order to accurately generate call volume, two related but separately collected data samples are required. Actual call volume is dictated by call arrival rates and call

durations. These two variables determine how often calls arrive in the system, and how long they remain in the system.

For the purposes of the TELUS project, data samples were used from the non-toll local calling networks of metropolitan Vancouver, Calgary and Edmonton. The most extensive and representative data sample, which forms the basis for the analysis in this report, comes from a detailed one-week recording of the TELUS non-toll local calling network for the Edmonton metropolitan area between May 6 and May 12, 2001. The Edmonton network consists of 12 backbone switches. Since calls may originate and terminate at any pair of different switches, the data can be broken down into traffic statistics for each of 132 different origin-destination (OD) pairs.

Data was collected in the form of "call-detail records". Key information in the sample includes where the call originated, where it terminated, which tandem switches were used (if any), what day and time the call was placed, and how long the call lasted. This information for a 1-week period resulted in a six gigabyte data set with more than fifty million records.

There are many underlying factors causing each of these OD-pairs to exhibit different call traffic characteristics; however, the most obvious factor is the mixture of residential and business customers within the origin and destination switch geographical areas. Figure 4-1 shows call volumes, which are measured in discrete bundles of 100 seconds of call time called centi-call seconds (CCS), for a pair of business switches by hour of day. Conversely, Figure 4-2 shows the same data for residential switches. Note that an observation on the graph indicates how much calling occurred between the two switches

during an hour. A measure of 10,000 CCS represents 1 million call seconds, the equivalent of approximately 278 calls lasting for the entire hour.

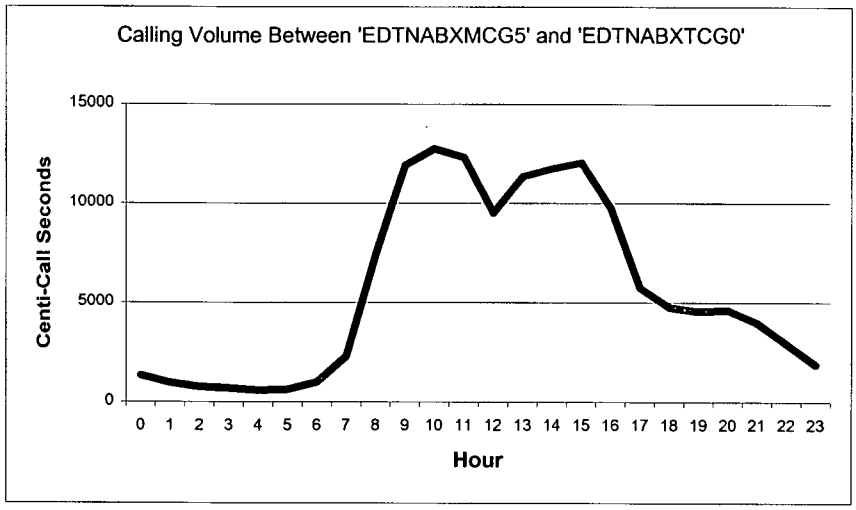


Figure 4-1: Call Volumes on a Business-to-Business Switch Pair measured by CCS

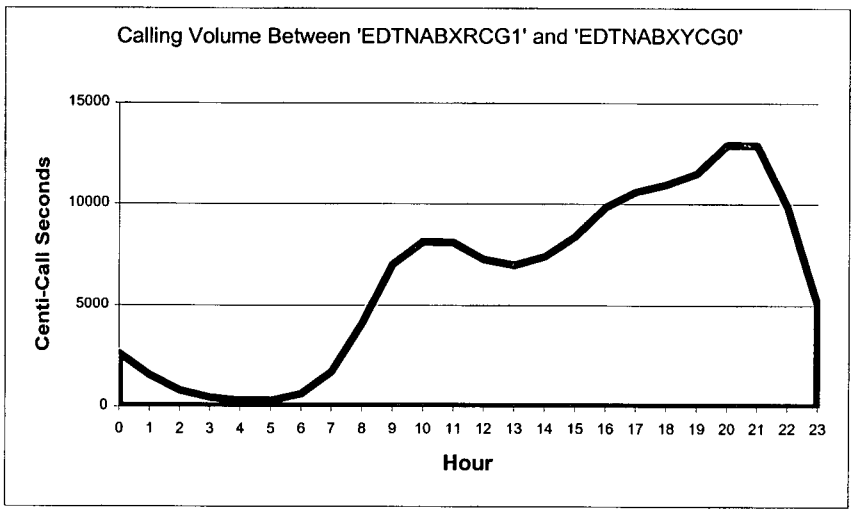


Figure 4-2: Call Volumes on a Residential-to-Residential Switch Pair measured by CCS

It is generally accepted that trunks with primarily business calling traffic are dominated by shorter duration calls occurring earlier in the day. In fact, the highest activity periods for business calling have traditionally been centered on 11:00 and 14:00 respectively as shown

in Figure 4-1. In contrast, trunk groups with primarily residential calling traffic are dominated by longer duration calls occurring much later in the day, since most individuals work during the day and are at their residences in the evening, again, Figure 4-2 appears to support this claim. While these two main factors dictate a large portion of the traffic characteristics for any OD-pair, there are certainly other factors at work as well, many of which will affect potential choices of distributions. These additional factors will be addressed in more detailed below. For the duration of this paper, I will refer to the switch pair 'EDTNABXMCG5' – 'EDTNABXTCG0' as a business switch, and 'EDTNABXRCG1' – 'EDTNABXYCG0' as a residential switch. Smith (2001) explored this distinction to propose "time-of-day" routing rules to enhance system performance.

4.1. Call Arrivals

Call arrivals are measured by rates. For any given origin-destination pair in the network, there is a certain rate at which calls arrive. In reality of course, this rate is not a deterministic steady stream of evenly spaced calls, but rather a distribution around a mean arrival rate. Theoretically, call arrivals should follow a non-stationary Poisson distribution.

4.1.1. Source explanation

Call arrival rates were calculated using call detail records from the Edmonton network. While interarrival times were not specifically tracked, call start times were available. Therefore, interarrival times were calculated by ordering calls according to start time and calculating differences. The data was binned by hour since the arrival process was

relatively stationary during any given hour. As a result, the first observation in each hour was discarded since no interarrival time could logically be computed. Interarrival rates were computed independently for each of 132 OD-pairs, during each hour of data collection throughout the week, resulting in 22,176 separate data series.

4.1.2. Graphical Summary

Figures 4-3 and 4-4 are histograms of total call arrivals by hour for the two switch pairs referred to above. Note that the binning of this data by hour is somewhat arbitrary, but is supported by the relative stationarity of intra-hour arrival rate data.

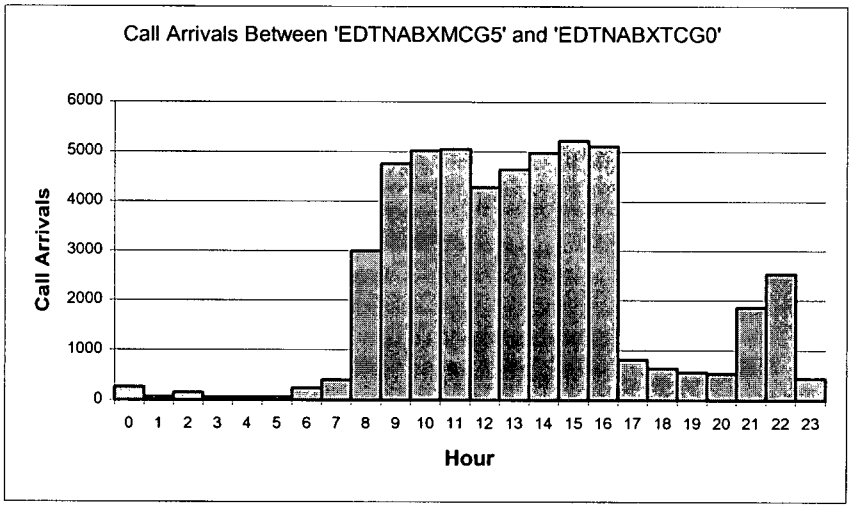


Figure 4-3: Call Arrivals on a Business-to-Business Switch Pair

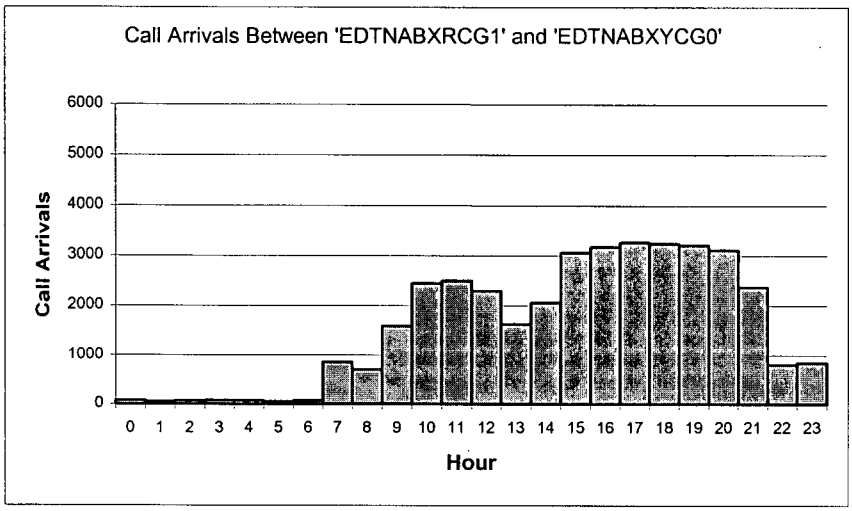


Figure 4-4: Call Arrivals on a Residential-to-Residential Switch Pair

It is interesting to note that the call arrival behavior is not unlike the total call volume on each switch pair. For business switches, the arrival rates climb to a mid-morning peak, fall during lunch, climb to an early afternoon peak, and fall off dramatically at the close of the business day. However, the residential switch pairs arrival rates do not mimic their total call volume profile in the same striking manner. Notice that the scales in Figure 4-3 and Figure 4-4 are the same. These graphs, along with the previous ones, suggest that residential switch pairs exhibit considerably lower calling rates than business switch pairs, even when the total observed volume is similar. A natural conclusion would be that the length of calls being placed differs; this is examined further in the next section.

4.1.3. Descriptive Statistics

Note that smaller mean interarrival times translate to greater arrival rates. Table 4-1 gives descriptive statistics for the interarrival data collected on our business-to-business (MCG5 – TCG0) and residential-to-residential (RCG1 – YCG0) switch pairs:

Table 4-1: Descriptive statistics of call interarrival times (in seconds)

	Business Switch			Residential Switch				99% CI μ_1- μ_2	
Hr	n_1	Mean(μ_1)	StDev	n_2	Mean(μ_2)	StDev	t-stat[3]	Lower – Upper	
00	732	4.89	7.07	85	41.76	51.52	6.59	22.43	51.32
01	745	4.81	7.15	41	82.83	92.26	5.41	40.81	115.23
02	700	5.13	7.77	39	81.43	71.22	6.69	46.83	105.75
03	685	5.25	7.75	20	135.84	146.57	3.98	45.93	215.24
04	736	4.88	7.26	49	68.13	99.39	4.45	26.58	99.92
05	833	4.31	6.00	171	20.56	23.22	9.09	11.64	20.87
06	1175	3.06	3.90	539	6.64	6.74	11.49	2.78	4.39
07	3006	1.20	1.29	1096	3.28	3.36	19.97	1.81	2.35
08	4137	0.87	0.85	1815	1.98	1.89	24.07	0.99	1.23
09	4196	0.86	0.84	2219	1.62	1.55	21.61	0.67	0.86
10	4033	0.89	0.85	2366	1.52	1.47	18.99	0.54	0.71
11	3338	1.08	1.06	2402	1.50	1.46	12.01	0.33	0.51
12	3846	0.94	0.91	2145	1.68	1.61	19.64	0.65	0.84
13	3871	0.93	0.89	2130	1.69	1.69	19.33	0.66	0.86
14	4057	0.89	0.86	2749	1.31	1.32	14.73	0.35	0.50
15	3196	1.13	1.09	3136	1.15	1.06	**0.81**	0.00	0.09
16	2038	1.77	1.86	3476	1.04	1.03	16.30	0.62	0.85
17	1646	2.19	2.38	3269	1.10	1.09	17.60	0.93	1.24
18	1571	2.29	2.33	2932	1.23	1.18	16.93	0.90	1.22
19	1494	2.40	2.39	2613	1.37	1.29	15.41	0.86	1.20
20	1111	3.24	3.75	1965	1.83	1.78	11.77	1.10	1.71
21	884	4.07	4.58	1084	3.31	3.62	3.99	0.27	1.24
22	782	4.59	5.15	394	9.10	10.29	8.22	3.10	5.94
23	768	4.66	5.31	162	22.20	21.80	10.18	13.09	21.98

It can be seen in Table 4-1 that business and residential interarrival rates are different at all time periods except one. However, even this similar time-period appears to be only the intersection of different trends. During the business peak hour, 10:00, this difference is 0.63 ± 0.085 seconds. When transformed to an arrival rate, this difference becomes 1.4 to 1.8 calls per second. Similarly, during the residential peak hour, 20:00, this difference is

[3] Mean differences between business and residential switches are highly statistically significant (T-test) for all hours except 15:00 – 16:00.

1.41 ± 0.305 seconds. When transformed to an arrival rate, this difference becomes 0.6 to 0.9 calls per second.

4.2. Call Durations

Call duration data represents the focus of this research, because it turned out to be the most difficult quantity to estimate and had the largest impact on the run time of the simulation. Unlike arrivals, which are measured by rates or interarrival times, call durations are measured strictly by elapsed time. For any given origin-destination pair in the network, there is a particular call duration pattern. Many things, including the level of business and residential traffic, as well as the nature of the calls themselves, govern this pattern. After all, one would imagine that a call placed to speak with a friend would differ in length from a call placed to make reservations at a local restaurant. In the past, these differences largely cancelled each other out and added up to an approximate exponential distribution for call durations. However, new types of calls are challenging this traditional duration pattern. More and more data calls are being placed on the circuit-switched voice network. These calls include modem calls to internet service providers, which typically last much longer than an average voice call. In addition, many calls are placed by merchants verifying credit card or debit card purchases. These calls should be very short in duration. While one might surmise that these calls would also serve to cancel each other's effects, it is useful to examine how well the traditional models fit the data. I will explore this issue in Section 5.

4.2.1. Data Source Explanation

Call durations were available from the call detail records provided by TELUS for the Edmonton network. Again, the data was arbitrarily binned by hour; however, in this case it was not for the purpose of generating a standard rate, but rather to examine some of the descriptive statistics and features of the data.

4.2.2. Graphical Summary

Following are histograms of mean call durations by hour for our two switch pairs of interest. Again, the binning of this data by hour is somewhat arbitrary.

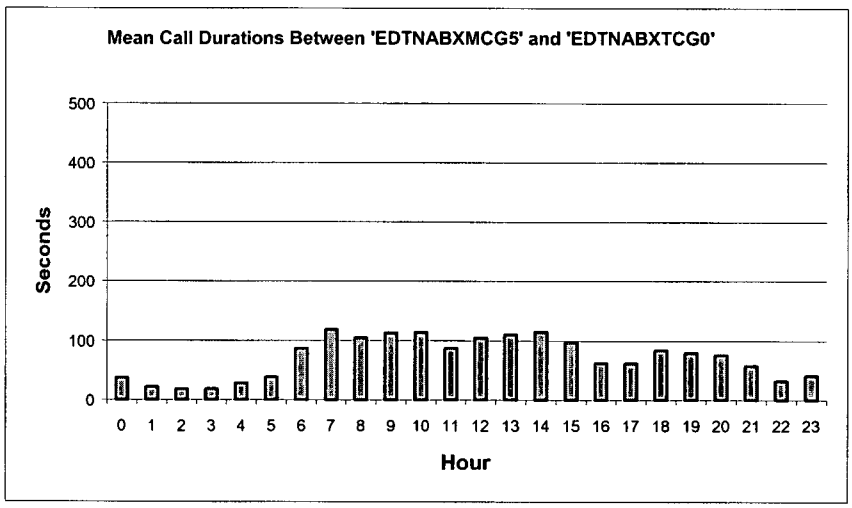


Figure 4-5: Mean Call Durations on a Business-to-Business Switch Pair

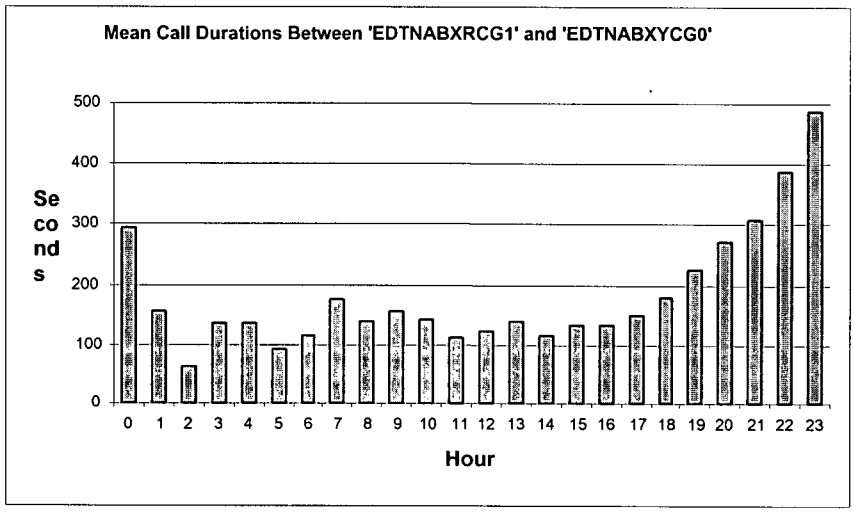


Figure 4-6: Mean Call Durations on a Residential-to-Residential Switch Pair

The graphs in Figure 4-5 and Figure 4-6 show that while calling volumes may be similar at peak times, the actual demand constituents may be radically different. These graphs both use the same scale, illustrating the dramatic difference in mean call duration between the business and residential peaks. This data, together with the call arrival patterns suggested by Figure 4-3 and Figure 4-4 make up the total call volume.

4.2.3. Descriptive Statistics

In examining differences between call durations, we notice that standard deviations are very high compared to the means. If the data were to fit traditional exponential models, we would expect to see mean and standard deviations nearly equal. Following are descriptive statistics for the duration data collected on our business-to-business (MCG5 – TCG0) and residential-to-residential (RCG1 – YCG0) switch pairs:

Table 4-2: Descriptive statistics for comparing mean call durations

	Business Switch			Residential Switch					99% CI μ_1- μ_2	
Hr	n_1	Mean(μ_1)	StDev	n_2	Mean(μ_2)	StDev	t-stat	Sig[4]	Lower – Upper	
00	732	37.42	234.06	85	293.73	960.54	2.45	*	48.42	464.21
01	745	21.67	209.08	41	155.94	460.20	1.86		-11.81	280.35
02	700	18.08	158.70	39	64.08	90.43	2.93	**	14.27	77.73
03	685	18.72	156.10	20	135.80	370.22	1.41		-56.63	290.80
04	736	28.42	263.55	49	135.82	276.98	2.64	*	25.49	189.33
05	833	38.60	181.89	171	93.76	191.36	3.46	***	23.72	86.62
06	1175	86.64	683.36	539	116.74	276.67	1.30		-15.53	75.72
07	3006	118.05	663.26	1096	174.44	397.09	3.31	***	22.96	89.81
08	4137	104.91	206.15	1815	138.13	314.64	4.13	***	17.43	49.01
09	4196	112.60	311.96	2219	155.72	369.90	4.68	***	25.05	61.18
10	4033	113.48	478.48	2366	141.65	340.26	2.74	**	8.01	48.33
11	3338	87.45	270.85	2402	111.46	301.70	3.10	**	8.84	39.18
12	3846	105.00	252.72	2145	123.05	302.92	2.34	*	2.93	33.16
13	3871	110.50	349.46	2130	139.73	328.52	3.22	**	11.44	47.01
14	4057	114.55	445.83	2749	117.08	277.55	0.29		-14.67	19.74
15	3196	97.10	228.96	3136	132.67	315.34	5.13	***	21.97	49.17
16	2038	63.00	204.90	3476	132.99	330.57	9.70	***	55.84	84.13
17	1646	62.25	234.52	3269	149.97	382.64	9.92	***	70.37	105.06
18	1571	84.29	330.13	2932	180.37	482.03	7.88	***	72.16	119.99
19	1494	80.10	376.46	2613	225.66	555.63	9.97	***	116.94	174.19
20	1111	76.18	495.80	1965	272.70	674.73	9.23	***	154.76	238.27
21	884	57.70	305.88	1084	308.47	831.34	9.20	***	197.26	304.28
22	782	31.86	132.51	394	387.16	1100.39	6.39	***	245.91	464.68
23	768	41.13	181.65	162	485.40	1675.73	3.37	***	183.95	704.59

The figures in Table 4-2 show that during the business and residential peak periods, call durations can vary significantly. During the morning peak hour, 10:00, this difference is only 28.2 ± 20.1 seconds, resulting in a gap between 8 and 48 seconds. This difference grows considerably throughout the remainder of the day, reaching 196.5 ± 41.8 seconds during the residential peak hour, 20:00, resulting in a gap between 155 and 238 seconds.

[4] Statistical significance of T-Test for different means at .05 (*), .01 (**), or .001 (***)

5. Call Demand Models

As was previously discussed, call demand should be modeled from the point of view of its two constituent components. As such, models were developed to fit distributions to both interarrival times and call durations. Both are presented, along with an analysis of their goodness of fit utilizing the criteria set forth in Section 2.

5.1. Call Arrivals

Under our hypothesis that call counts should follow a Poisson Process, we began this phase of the project by attempting to fit exponential distributions to interarrival times. If interarrival times are fit well by an exponential distribution with some mean $1/\lambda$, then it follows that the call arrivals will follow a Poisson Process with parameter λ. However, while a useful property, the fact that the arrivals follow a Poisson Process is not requisite when designing a simulation. In a discrete-event based simulation the only random variable being simulated is the interarrival time. If we can find a significantly better fit using a model of mixed exponential distributions, we would no longer have a Poisson Process, but we would not add much complexity to the process of generating interarrival times, since generating data from a mixed exponential random variable is not a complex operation.

5.1.1. Fitting A Single Exponential Distribution to Interarrival Rates

Fitting a single exponential distribution is a fairly straightforward process. Once the data is in the requisite format (interarrival times calculated by differences in start times), a

histogram can be created, and compared with a theoretical distribution by way of the SAS procedure "PROC CAPABILITY[5]." This SAS procedure fits a single exponential distribution to the indicated data using maximum likelihood estimation. The procedure is performed by SAS, and the complete output is shown in Appendix 1. For our two example switch pairs, Business switches at 10:00, and Residential switches at 20:00, the procedure was run and test statistics were generated.

Table 5-1: Goodness of fit statistics for Call Arrivals (Full Data Set)

Switch Type	Mean Interarrival Time	Chi-Squared Test		Kolmogorov-Smirnov Test	
		Statistic	p-value	Statistic	p-value
Business	0.891718	165.003	<0.001	0.106	<0.001
Residential	1.830534	25.314	.046	0.054	<0.001

Notice in Table 5-1 the MLE procedure estimates the mean of the exponential distributions at 0.892 seconds for the Business switches, and 1.831 seconds for the residential switches. However, both tests indicate that we should reject the null hypothesis in either the business or residential case. Recall, that the null hypothesis in both of these tests is that the theoretical distribution is equal to the "true" data-generating distribution. It would appear then that this fitting procedure did not find an appropriate theoretical distribution. However, as was mentioned before, statistical goodness of fit tests can be problematic when examining particularly large data sets.

[5] For more information about PROC CAPABILITY, see SAS user guide.

In this case, the business switch sample consists of 4033 observations, while the residential switch example has a sample size of 1965 observations. In order to examine the data for practical significance, rather than statistical significance, we can try testing smaller samples of the data against the theoretical distribution. This is accomplished by taking random samples of the data, and recomputing the test-statistic.

Again, SAS was used to take smaller random samples of the data. For each of these smaller data sets, an empirical distribution was calculated, and a Kolmogorov-Smirnov statistic calculated by examining deviation from the theoretical distribution indicated in Table 5-1. Using the smaller data sample, the critical value of the Kolmogorov-Smirnov statistic becomes 0.215034 at the .05 significance level.

Table 5-2: Goodness of fit statistics for Call Arrivals (Smaller Random Samples)

Type	Number of Samples	Critical Value at .05 significance	Number above critical value	Number below critical value
Business	100	.215035	5	95
Residential	100	.215035	4	96

These results indicate that of these smaller samples, only approximately 5% of the observations lie beyond the .05 significance critical value. While this manner of testing is not as powerful as examining the entire data set, it can be used to establish the practical validity of the model. Further testing by way of simulation is certainly warranted.

Another way of comparing the empirical distributions of call arrivals with their maximum likelihood exponential distributions is visually. Figure 5-1 shows a probability plot for the empirical distribution of call interarrival times on the business switches, and the corresponding exponential distribution suggested by maximum likelihood estimation. Assuming the data did come from the theoretical distribution, we would expect it to fall in

a linear pattern directly on the theoretical line. In Figure 5-1 we see that after approximately the 90th percentile, the data begins to exhibit characteristics of a heavier tail than the theoretical distribution predicts. This deviation from linearity is small, and does not occur until the extreme points of the distribution. Also, the number of observations occurring with larger deviations is quite small. Practically, this deviation results in somewhat longer interarrival times, as a result, when simulating, a simple exponential distribution may produce more calls than the "true" distribution, thus overstating overall demand on the network.

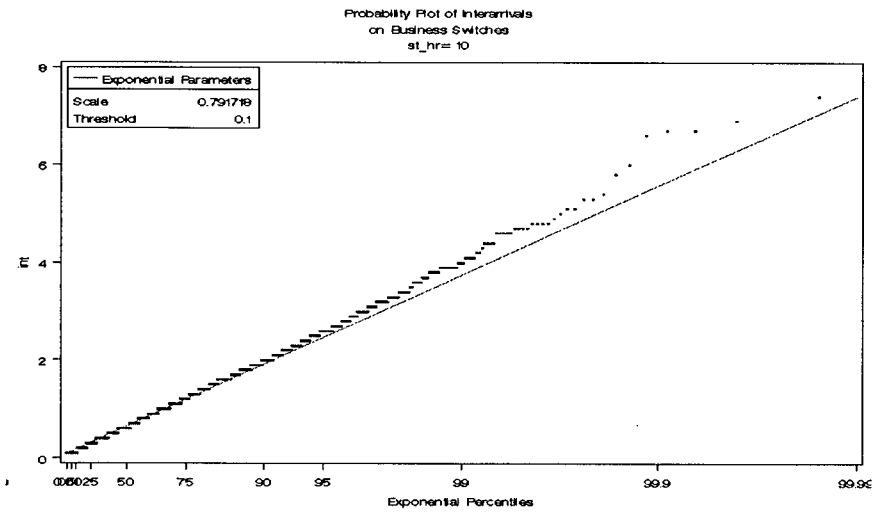


Figure 5-1: Probability Plot of Interarrival times on Business switches

We can see similar behavior of the call interarrival times on residential switches in Figure 5-2. However, in this case, notice that there are extreme observations both above and below the expected distribution. We can be somewhat more confident using an exponential distribution to model the residential switch interarrivals then in the case of business switch calls. However, both appear to exhibit exponential behavior, as confirmed by the goodness of fit tests performed above.

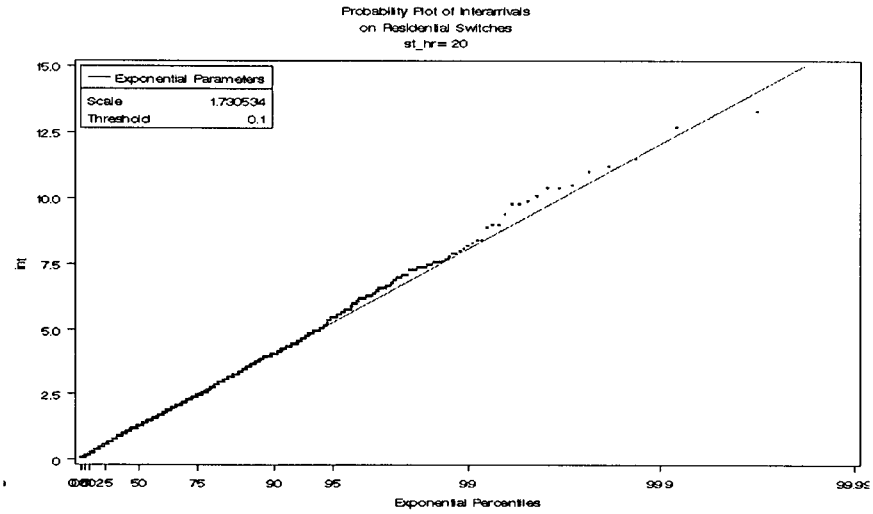


Figure 5-2: Probability Plot of Interarrival times on Residential switches

5.1.2. Fitting A Mixed Exponential Distribution to Interarrival Rates

As discussed earlier, the arrivals following a Poisson Process is useful for analytical evaluation of system performance, but is not necessary, even for ease of simulation. Accordingly, tests were performed to fit a mixed exponential distribution. Recall that a mixed exponential probability density with K component densities takes the following form:

$$g(x) = \sum_{j=1}^{K} \pi_j \cdot \lambda_j e^{-\lambda x}$$

and, has the following constraint:

$$\sum_{j=1}^{K} \pi_j = 1$$

Fitting this more complicated distribution to the observed data proves more difficult than fitting a single exponential density. While SAS is able to compute maximum likelihood estimates of some basic probability distributions, it is ill equipped to deal with mixtures.

As a result, code was written specifically for the purpose of solving the maximum likelihood estimates optimization problem[6] where the mixture density formula is used as the objective function. The model was programmed using the "PROC NLP" procedure (Non-linear programming), and detailed code can be found in Appendix 2. Since we are using a mixture of only two exponential distributions ($K = 2$), there are only three parameters: $\pi, \lambda_1, \lambda_2$. As indicated above, we do not need to solve for a second mixing parameter, since this value will simply be $1 - \pi$.

Table 5-3: Maximum likelihood estimates for mixed exponential interarrivals

Type	π	λ_1	λ_2
Business	-3.7×10^{-18}	3.6619	0.891719
Residential	2.9×10^{-19}	6160.025	1.830534

The values produced by the fitting procedure are shown in Table 5-3. Observe that the value of π in both cases is effectively zero. As a result, we can ignore λ_1, since the procedure is in fact recommending that a mixture is not necessary, as it cannot provide a better fit to the data. Logically then, we find that the estimates for λ_2 are in fact equivalent to the estimates for our single exponential as shown in Table 5-1, with the exception of a deviation of 0.000001 on the business switches which is likely due to rounding error.

Obviously then, nothing is gained by mixing together two exponential distributions to account for interarrival times. We can conclude that interarrival times can be adequately

---

[6] See Section 3.1 for formulation of this optimization problem.

modeled using a single exponential distribution. Since there is effectively no change to the input parameters, there is similarly no change to the goodness of fit statistics.

5.2. Call Durations

The process of modeling call durations proves to be more difficult than call arrivals. In order to illustrate the problem, I will examine the model from the perspective of the two previously chosen data sets. As already shown in Figure 4-5 and Figure 4-6, we can expect residential calls to have longer mean durations than business calls. However, what about the constituent components which make up this distribution of call durations. Can we expect a more or less variable distribution between these switch pairs? As with call arrival modeling, having a simple probability distribution is not necessary in order to generate simulations. Regardless of its complexity, provided a closed form version of the cumulative density function can be written, it can easily be used in our discrete event simulation. Accordingly, I will seek out a distribution function which describes call durations accurately.

5.2.1. Fitting A Single Exponential Distribution to Call Durations

As with call arrivals, fitting a single exponential distribution for call durations is not difficult. SAS procedures may be used to generate a histogram of the relevant data, fit a simple theoretical distribution, and produce goodness of fit statistics. As with the procedure for call arrivals, processing is done internally using the "PROC CAPABILITY" procedure in SAS. The complete output from this procedure can be found in Appendix 1.

The procedure was run on the two example switch pairs, business and residential, and test statistics were generated.

Table 5-4: Goodness of fit statistics for Call Durations (Full Data Set)

Switch Type	Mean Duration	Chi-Squared Test		Kolmogorov-Smirnov Test	
		Statistic	p-value	Statistic	p-value
Business	0.010977	455652.452	<.001	0.333	<.001
Residential	0.006166	3106623.90	<.001	0.30	<.001

The results of this analysis indicate that an exponential distribution which maximizes the probability density of the given dataset has a mean of 91.1 seconds for business switches, and 162.19 seconds for residential switches. This validates our interpretation that residential users talk longer, but place fewer calls, given the visual data in Section 4. As was the case with fitting single exponential distributions to large samples of interarrival times, Table 5-4 shows that the procedure has produced test statistics which indicate we should reject the null hypothesis. Again, we can understand that this rejection is based largely upon the fact that the data size used is so large. In this case, the data set for call durations contains 49,580 observations for the business switches, and 36,897 observations for the residential switches, both of these data sets make it extremely difficult to statistically accept a null hypothesis.

As with our investigations of the call arrival distributions, it is certainly possible that this large number of observations is masking a practically useful estimate for the distribution. In order to investigate this, we shall again bootstrap by taking random samples from the data and testing them against the theoretical exponential distributions, as well as plotting probability plots to look for visual clues.

SAS was used to take random samples of the data. Within each of these samples, an EDF was computed and compared to the fitted exponential distribution to produce a Kolmogorov-Smirnov test statistic.

Table 5-5: Goodness of fit statistics for Call Durations (Smaller Random Samples)

Type	Number of Samples	Critical Value at .05 significance	Number above critical value	Number below critical value
Business	1000	.215035	1000	0
Residential	1000	.215035	1000	0

Unfortunately, in this situation, our bootstrapping technique has served only to reinforce the initial suggestion to reject the null hypothesis. Table 5-5 shows that all 1000 random samples taking from the data fail to meet the critical value of the Kolmogorov-Smirnov test statistic even with this much lower number of observations. The evidence is clear that a simple exponential distribution may not adequately describe the observed data. In order to confirm this, we can look at a probability plot of the data.

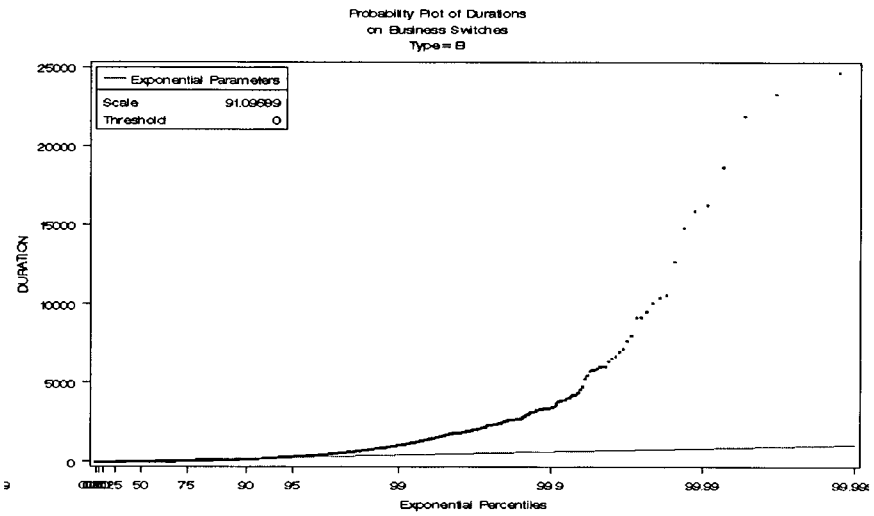


Figure 5-3: Probability Plot of call durations on Business switches

Figure 5-3 clearly confirms our suspicions. While the data from the business switches appears to follow the exponential pattern up until the 95th percentile, it then begins to deviate from the theoretical distribution in an extreme manner. This phenomenon is described as the distribution having "fat tails", since its variance is obviously much higher than the estimating distribution's. This graph can be compared with Figure 5-2 for the call interarrival times, showing a data set which clearly does follow the exponential distribution. Similarly, we can see that the same observations hold for the data from the residential switches. Figure 5-4 shows a similar pattern, with somewhat less extreme variation, suggesting that the residential call durations are closer to exponential than business call durations, though the exponential model is still grossly inadequate.

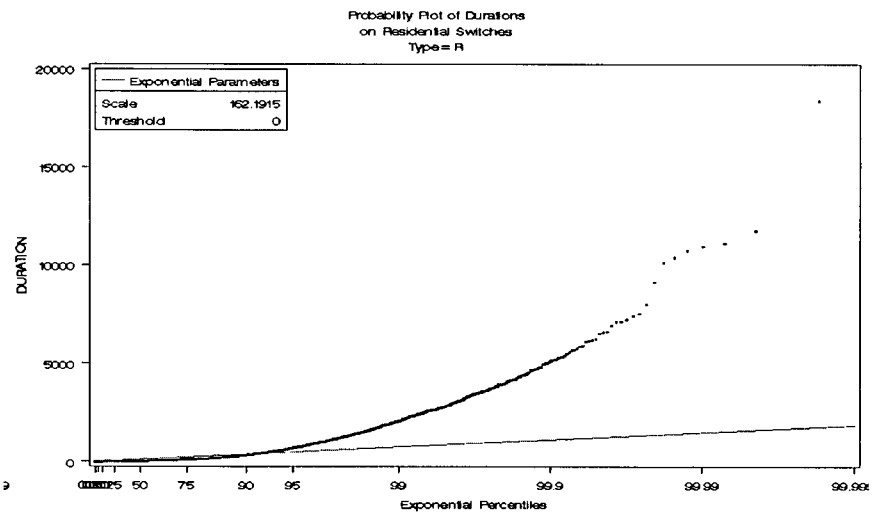


Figure 5-4: Probability Plot of call durations on Residential switches

5.2.2. Fitting A Mixed Exponential Distribution to Call Durations

As with the call arrival distributions, we can fit a mixture of exponential distributions to the data in order to capture multiple levels of variability. Fitting this mixture of distributions has great intuitive appeal when dealing with call durations. After all, the

classical view that telephone calls follow an exponential distribution has held for many years; it has only begun to change recently as calling demands have begun to change. In fact, it is still intuitively reasonable to suppose that all calling traffic of the same type should follow an exponential distribution. For example, voice traffic should follow an exponential distribution with some mean, while data calls should follow another exponential distribution with some other mean; and so on, through credit-card verifications; calls terminating through call-waiting or answering machines; and faxes. Since our dataset does not label each call according to these properties, we are forced to try to derive not only what these various distributions are, but also how many of them we should use.

In order to establish what these distributions are, we will use two techniques. The first technique is non-linear programming using SAS as was described in calculating mixtures for call arrivals. Detailed code used for this method can be found in Appendix 2. The second method used is the EM algorithm which was discussed from a theoretical point of view in Section 3.2. The detailed programming instructions used for this method are in C++ and can be found in Appendix 3.

As has been well documented in the literature, it is difficult to determine how many component densities are appropriate to use. Consequently, and since we know this number should be relatively small, we have fit distributions first using two component densities, and then one more sequentially, testing significance at each step until no further significant gains can be made.

The detailed results of each iteration of the algorithms is presented for the two-component density mixture in Appendix 4. It is interesting to note that running times for the two methods were quite similar, with both generally producing parameter estimates for a given switch pair in less than a minute.

Table 5-6: Parameter estimates for a two-component mixture density

Switch Type	NLP Estimates	EM Estimates
Business	$\pi_1 = 0.6595$ $\lambda_1 = 0.0072$ $\pi_2 = 0.3405$ $\lambda_2 = 3.4069$	$\pi_1 = 0.6597$ $\lambda_1 = 0.0074$ $\pi_2 = 0.3403$ $\lambda_2 = 3.4077$
Residential	$\pi_1 = 0.1963$ $\lambda_1 = 0.0016$ $\pi_2 = 0.8037$ $\lambda_2 = 0.0199$	$\pi_1 = 0.1963$ $\lambda_1 = 0.0016$ $\pi_2 = 0.8037$ $\lambda_2 = 0.0199$

Table 5-6 shows that both methods lead to the same parameter estimates for the residential switches, and have only minor differences in the estimation of means for the business switches. For our purposes, we can assume these are equal, since data was collected only to the tenth of a second level of detail.

The parameters generated by these methods have some interesting properties. Since the mean of the exponential is the reciprocal of the rate parameter, we observe that for business switches, these methods yield a mixture of exponentials with means of 138 and 0.29 seconds. For residential switches, the method equates to mixing together exponentials using means of 50 and 625 seconds. There is probably ample reason to explain this. After all, most merchant credit card verification calls (abnormally short) would originate from business switches, while most internet calls (abnormally long) would originate from residential switches.

A look at Appendix 2 and Appendix 3 reveals that programming and using the NLP code is considerably easier than the comparable effort required to use the EM Algorithm. This is largely because many "off-the-shelf" software packages support some level of non-linear programming, while any implementation of the EM Algorithm will necessarily require some amount of customized programming. Given that computation is much easier using technology available today, perhaps the longer solution times are acceptable given the reduction in implementation complexity.

Returning to our two-component mixture density, the next step is to test it against the data and see if this distribution provides a significantly better fit than the single exponential distribution did. In order to make this comparison, I will compare Kolmogorov-Smirnov test statistics from random samples of the data.

Table 5-7: Goodness of fit statistics for Two-Component Call Durations

Type	Number of Samples	Critical Value at .05 significance	Number above critical value	Number below critical value
Business	1000	.215035	275	725
Residential	1000	.215035	52	948

The goodness of fit statistics shown in Table 5-7 indicate that the two-component mixture density is a better description of the data than the single exponential distribution examined in Table 5-5, in which all of the random samples suggested we reject the null hypothesis. In this situation, only 27.5% of the random samples suggest rejecting the null hypothesis for the business switches, while 5.2% of the random samples suggest rejecting the null hypothesis for the residential switches.

It appears then, that we may have found an adequate description of the residential switch data. After all, we would expect approximately 5% of random samples to exceed the critical value of the Kolmogorov-Smirnov statistic at the 95% confidence level. However, the model for the business switches could certainly be improved. Considering the obvious gains we have made by adding a second component density, we should try adding a third.

Table 5-8: Parameter estimates for a three-component mixture density

Switch Type	NLP Estimates		EM Estimates	
Business	$\pi_1 = 0.0638$	$\lambda_1 = 0.0014$	$\pi_1 = 0.0644$	$\lambda_1 = 0.0014$
	$\pi_2 = 0.5985$	$\lambda_2 = 0.0130$	$\pi_2 = 0.5980$	$\lambda_2 = 0.0131$
	$\pi_3 = 0.3377$	$\lambda_3 = 3.4225$	$\pi_3 = 0.3376$	$\lambda_3 = 3.4223$
Residential	$\pi_1 = 0.7226$	$\lambda_1 = 0.0229$	$\pi_1 = 0.7225$	$\lambda_1 = 0.0229$
	$\pi_2 = 0.2154$	$\lambda_2 = 0.0038$	$\pi_2 = 0.2154$	$\lambda_2 = 0.0038$
	$\pi_3 = 0.0620$	$\lambda_3 = 0.0008$	$\pi_3 = 0.0621$	$\lambda_3 = 0.0008$

These three-component mixture parameters shown in Table 5-8 differ somewhat from those in Table 5-6 for the two-component mixtures.

For business switches, the methods now suggest mixing together three exponentials, with means of 0.29, 77, and 714 seconds. Notice that the first mean has not changed, but the second one from Table 5-6 has been expanded to include both shorter (77 sec), and longer (714 sec) voice traffic. Notice also that the particularly long calls will only occur approximately 6% of the time.

Similarly, for residential switches, we now have means of 44, 263, and 1250 seconds. Again, the first mean has not changed drastically, while the second one has been extended to both shorter and longer calls.

Mathematically, the three component mixture should certainly be a better description of the data than the two-component, since the two-component is a special case of the three component density where one of the proportion parameters equals zero. However, it is interesting to compare the goodness of fit statistics in order to determine how much better this new density describes the data.

Table 5-9: Goodness of fit statistics for Three-Component Call Durations

Type	Number of Samples	Critical Value at 95% significance	Number above critical value	Number below critical value
Business	1000	.215035	117	883
Residential	1000	.215035	49	951

As we can see from the goodness of fit statistics, the residential calls did not gain much by adding the third component density. The number of samples exceeding the critical value at 95% significance is still our expected 5%. This holds with both our predictions after seeing the statistics for the two-component densities, and with our intuition that the three-component density might not be adding much value to the residential switches.

However, for the business switches, we see a dramatic change. The number of random samples exceeding the critical value at 95% confidence has dropped from 27.5% with two components, down to 11.7% with three. While this is promising, we would be remiss to not try fitting four components, in case further gains can be made.

We should have no expectation of making further gains from applying a four-component distribution to the residential switch data, since we have already found a mixture distribution which adequately describes this empirical data. The analysis is included for consistency, however.

Table 5-10: Parameter estimates for a four-component mixture density

Switch Type	NLP Estimates		EM Estimates	
Business	$\pi_1 = 0.3376$	$\lambda_1 = 3.4225$	$\pi_1 = 0.3091$	$\lambda_1 = 3.4225$
	$\pi_2 = 0.5985$	$\lambda_2 = 0.0130$	$\pi_2 = 0.5983$	$\lambda_2 = 0.0130$
	$\pi_3 = 0.0000$	$\lambda_3 = 0.0167$	$\pi_3 = 0.0287$	$\lambda_3 = 3.4225$
	$\pi_4 = 0.0638$	$\lambda_4 = 0.0014$	$\pi_4 = 0.0639$	$\lambda_4 = 0.0014$
Residential	$\pi_1 = 0.3613$	$\lambda_1 = 0.0229$	$\pi_1 = 0.6752$	$\lambda_1 = 0.0229$
	$\pi_2 = 0.3613$	$\lambda_2 = 0.0229$	$\pi_2 = 0.0474$	$\lambda_2 = 0.0229$
	$\pi_3 = 0.2154$	$\lambda_3 = 0.0038$	$\pi_3 = 0.2154$	$\lambda_3 = 0.0038$
	$\pi_4 = 0.0620$	$\lambda_4 = 0.0008$	$\pi_4 = 0.0620$	$\lambda_4 = 0.0008$

A quick glance at Table 5-10 may lead you to believe that a four-component mixture has made some improvement over the three component mixtures of Table 5-8. However, closer inspection will reveal that it has in fact not.

Beginning with the business switches, first note that the solutions from the NLP and EM algorithm are more similar than they appear. The non-linear program has found one component to be irrelevant ($\pi_3 = 0$). At the same time, the EM algorithm has allocate two components to the same underlying distribution ($\lambda_1 = \lambda_3 = 3.4225$). Accordingly, their relative proportion parameters can be combined to equal 0.3378. Notice that this value is nearly identical to the value of π_1 from the non-linear program. The resulting three component densities of these two manipulations are in fact nearly equal to those described in Table 5-8.

The residential switch distributions may be reduced in a similar fashion. Notice that the first two components from each technique can be reduced to a proportion parameter of

0.7226 and an exponential distribution parameter of 0.0229. Also, after reducing to these three parameters, the distributions become equal to those of Table 5-8.

Thus, we have shown that adding a fourth component distribution does not add any value to the procedure, since the fourth component is not able to account for any additional variation. Consequently, no goodness-of-fit statistics will be calculated on the four-component distributions, and no additional number of components will be fit to the empirical data. However, the three component distribution does not appear to adequately describe the duration of calls on business switches according to the Kolmogorov-Smirnov statistic.

No models were developed for time periods other than the busy hour indicated in Section 4.2, since the network is "over-engineered" (has excess capacity) with respect to all other time periods. The regulatory requirement for *p.01* quality of service is imposed only upon the "busiest" hour of the week. Since the purpose of the simulation was to enable impact analysis of capacity reduction decisions, only the duration and arrival distributions for the busiest hour were relevant.

5.2.3. Review of Fitted Models

Throughout Section 5.2, we have shown progressively more complicated models for depicting call duration data. It is useful now to review the distributions suggested by each model in a way which allows them to be compared. Two visual comparisons will be made:

1. Q-Q Plots of empirically observed quantiles vs. theoretically expected quantiles.
2. Boxplot comparisons of Kolmogorov-Smirnov statistic distributions

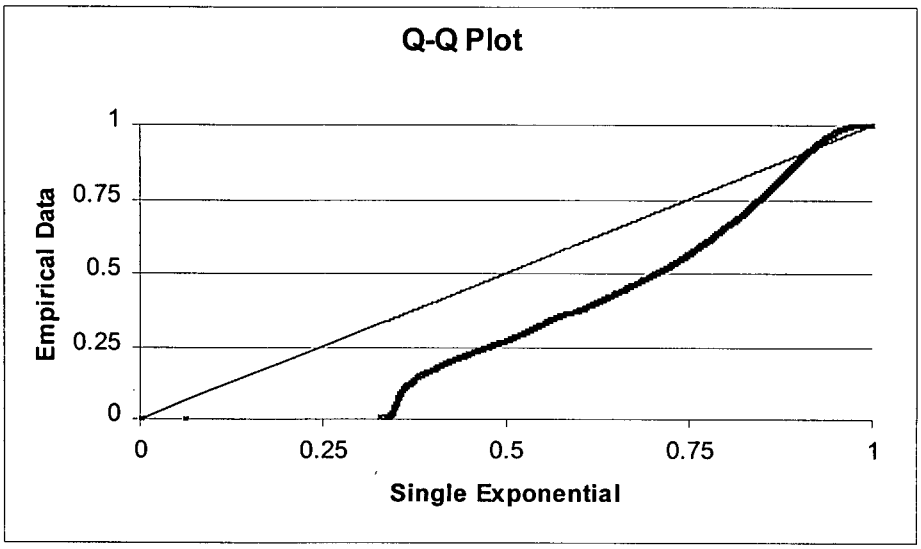


Figure 5-5: Analysis of Business Switches Single Exponential Distribution

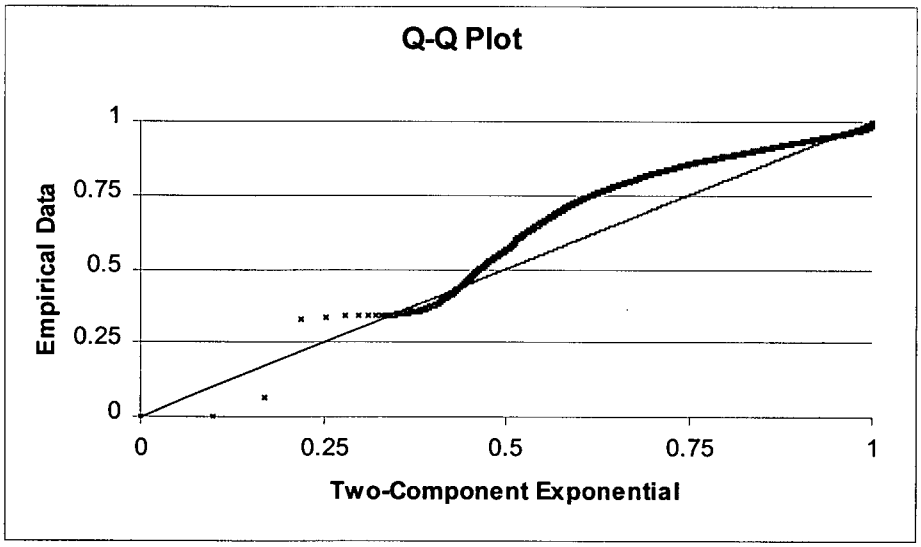


Figure 5-6: Analysis of Business Switches Two-Component Mixture Distribution

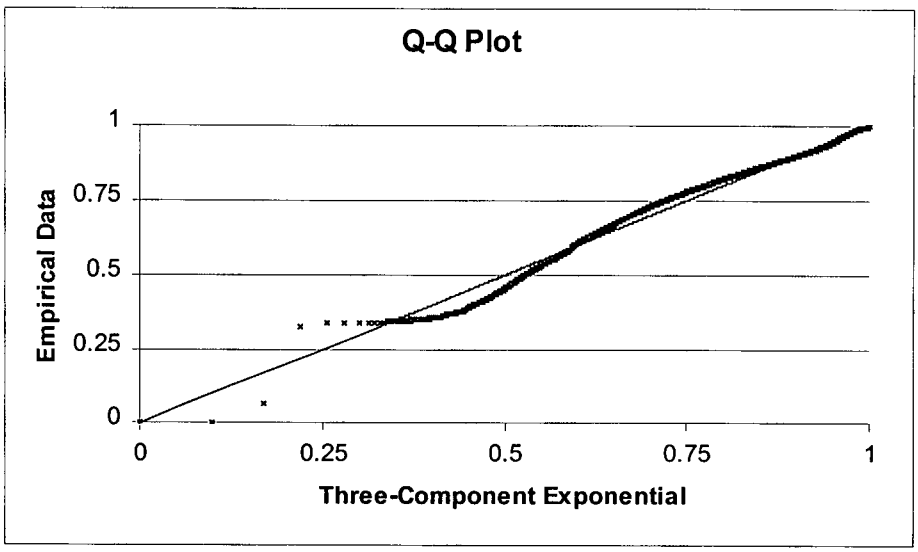


Figure 5-7: Analysis of Business Switches Three-Component Mixture Distribution

Figure 5-5 shows a Q-Q plot of the single exponential distribution. Note that the single distribution does not even come close to the reference line which indicates a perfect match with the observed empirical proportions. This is exactly what we expected from our goodness-of-fit tests. Figure 5-6 and Figure 5-7 show the same plots for two and three component exponential mixture distributions respectively. Note that both of these distributions are much closer to the reference line, and that the three-component density appears to fit the distribution best.

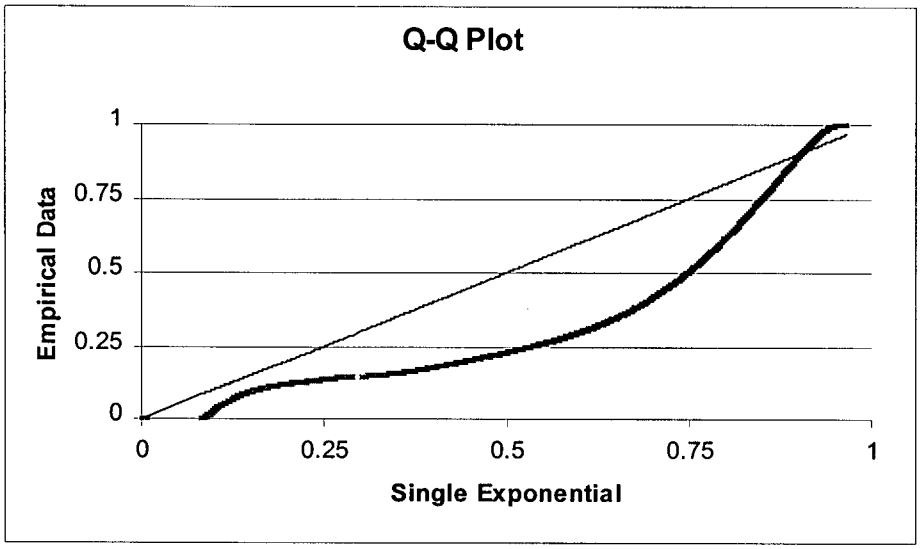


Figure 5-8: Analysis of Residential Switches Single Exponential Distribution

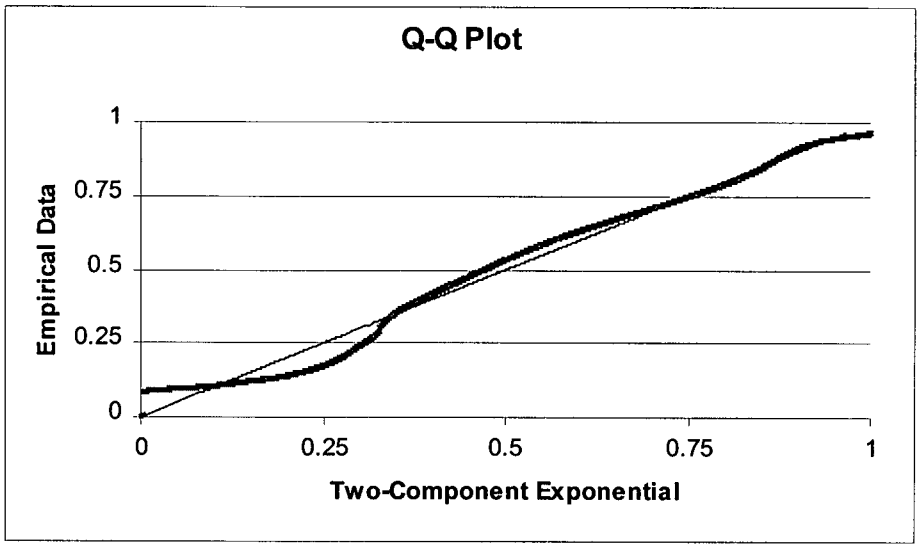


Figure 5-9: Analysis of Residential Switches Two-Component Mixture Distribution

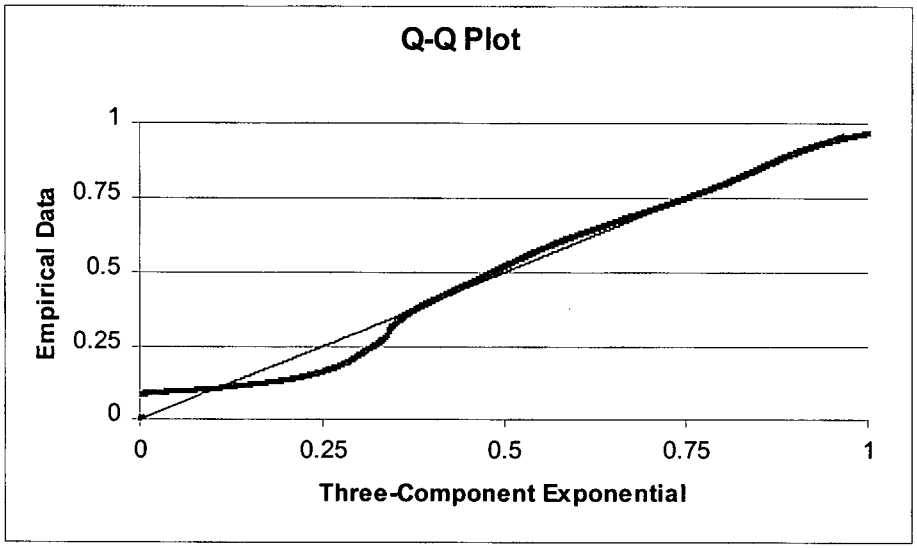


Figure 5-10: Analysis of Residential Switches Three-Component Mixture Distribution

As with the business switches, Figure 5-8 shows that a single exponential distribution is an inadequate fit for the empirically observed data. However, in this case, while we do see a dramatic improvement when moving to a two-component density (Figure 5-9), we do not see any marked improvement when moving to a three-component density (Figure 5-10). Again, this simply reinforces our conclusions from the goodness-of-fit tests, since we already observed that moving to a three-component density did not increase the suitability of the model.

When comparing the Kolmogorov-Smirnov tests earlier in the previous section, it is important to keep in mind that these test statistics were generated by a bootstrapping technique of repeatedly sampling from the data, in order to avoid the problems associated with fitting distributions to large data sets. Consequently, instead of a single Kolmogorov-Smirnov statistic, we in fact have a distribution of 1000 for each density type. These

distributions can be compared to see which mixture density is more likely to result in random samples which meet the Kolmogorov-Smirnov critical value.

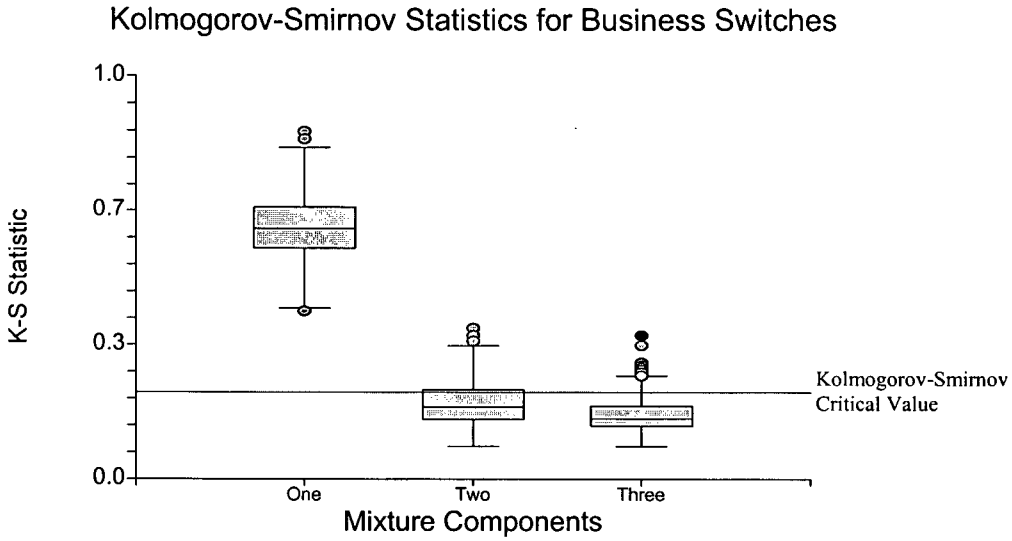


Figure 5-11: Boxplots of Kolmogorov-Smirnov Statistics for Business Switches

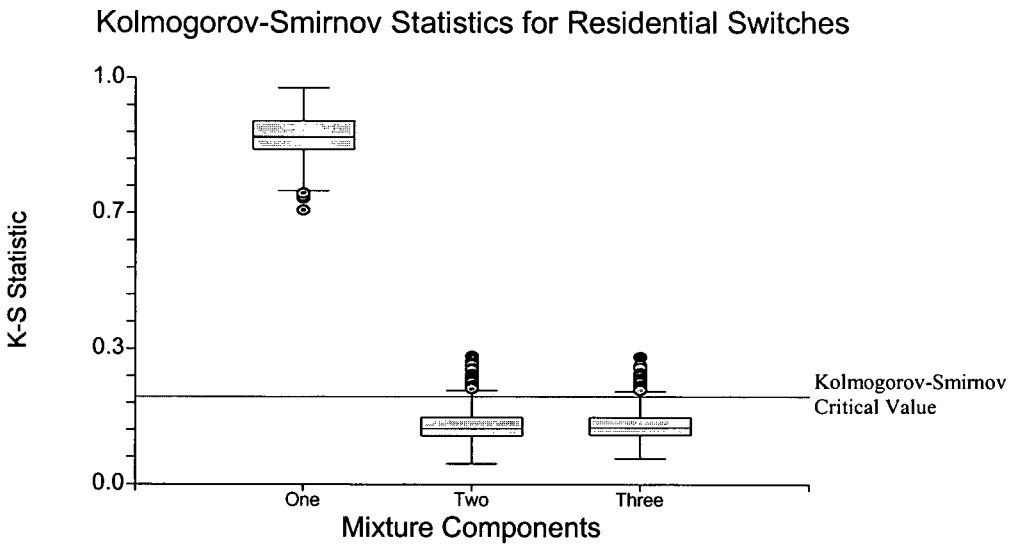


Figure 5-12: Boxplots of Kolmogorov-Smirnov Statistics for Residential Switches

Notice that both Figure 5-11 and Figure 5-12 agree with our earlier conclusions. Most of the Kolmogorov-Smirnov statistics lie below the critical value (shown as a horizontal line),

when more than two component densities are mixed together for both business and residential switches. Again, the entire distribution shifts lower when a third component is added for business switches, but not for residential switches.

6. Validation

As discussed in Section 2, more than one criteria can be used in order to evaluate the suitability of a candidate distribution function. While building the models, a number of statistical goodness-of-fit tests were used. However, in addition to these tests, one can measure the performance of the actual simulation when the new distribution functions are used, in order to ensure that proper performance is maintained, and to measure the impact upon decision-making this shift has caused.

For more information about the workings of the circuit-switched network simulator, please see Braun (1999). In addition to this work, the simulator has now been modified to use multiple component densities for call interarrival as well as call duration times.

6.1. Quantitative Validation

In order to validate the working of the new circuit-switched network simulator using the modified probability distributions, we run the new simulator and compare its output to our empirical data. Call detail records are created by the simulation in order to provide the number of calls placed and the duration of each call using the new probability distributions. These values are then compared to the statistics collected from the actual network in order to validate the use of these distributions. This procedure required that the

simulation be configured in a specialized manner and it took a great deal of time both to run the simulation and to compare the outputs.

Running the simulator with the empirical distribution inputs always yields call traffic outputs very similar to those specified in the input distributions. Switching over to mixture densities, we observe the deviation in call traffic according to two metrics and at four different time periods. The two metrics used are call arrivals and volume in centi-call seconds (CCS). Call arrivals is simply a count, meaning it is incremented by one each time a call is placed. CCS is a measure of overall volume of usage, combining the effects of arrivals and durations. This second measure is subject to random fluctuations in both arrival rates and durations, thus it is a good indicator of the overall effectiveness of using two theoretical distributions simultaneously. In addition, the CCS data is not used directly to parameterize the distributions, rather the constituent pieces (arrivals and durations) are used, thus CCS also provides a somewhat independent set of data to verify the simulation. We validate the distributions at four time periods simply because the simulator is set up to run single hours, and these four chosen hours represent all busy periods on the network.

Table 6-1: Difference between simulation output and empirical data for all switch pairs

	Call Attempts (Count)				Calling Volume (CCS)			
	Empirical	Mixture	Difference	% Diff.	Empirical	Mixture	Difference	% Diff.
Weekday Morning	409,542	407,495	2,047	0.5%	1,147,654	1,125,849	21,805	1.9%
Weekday Afternoon	418,539	432,351	-13,812	-3.3%	1,184,009	1,242,026	-58,016	-4.9%
Weekday Evening	302,876	299,242	3,634	1.2%	1,240,139	1,232,698	7,441	0.6%
Weekend Evening	51,978	51,926	52	0.1%	263,777	259,293	4,484	1.7%

Notice in Table 6-1 that the largest deviations occur during the same time period, the Weekday Afternoon. However, even this deviation is less than 5% of the expected observations. In addition, deviation from empirical results did not have any systematic bias, various replications yielded results above and below the expected values.

6.2. Impact on Decision-Making

One unintended impact which switching to theoretical distributions had upon the project, was the effect on the run time of the simulation. As indicated in Braun (1999), the simulator is coded in C++. Since the simulation deals with millions of virtual telephone calls over its hour-long horizon, it typically took as long as two hours to run. However, through the use of theoretical distributions, as well as a new indexing technique for the data files, this run time was decreased to approximately fifteen minutes. While not crucial to its implementation, this faster run-time allowed users easier access to the simulation and its results. Users are typically not used to dealing with applications which take much longer than a couple seconds to process their requests, so any reduction in the required time was welcomed by the user team, and our project team.

Another aspect of our project which is developed more thoroughly in Sim (2001), is the study referred at the bottom of Figure 1-2. The cost savings from reduced trunking were studied using a new type of call routing termed "alternate, time-of-day routing". Without diving into the domain of this separate research, I want to mention that the switch to mixed exponential distributions had some affect on our recommendations. First of all, this switch allowed us to run the simulation engine much faster, which was very valuable given that

the impact study contained eight different base case scenarios which had to be simulated and compared against each other. Second, the use of theoretical distributions gave us greater confidence in the fact that we were modeling something closer to the actual random variables than we had been while using the empirical distributions.

The result of this study which our new augmented simulation played a pivotal role in was that TELUS, and potentially other telecommunications organizations could save as much as three percent of their current trunking costs if they are currently using a direct-routed system, simply by switching over to "alternate, time-of-day routing".

7. Integration

Establishing the suitability of a particular probability distribution in describing a random variable is only the first step in implementing a required solution. Given a pre-functioning simulation engine, the next step in the process was to integrate these derived distributions with the pre-existing architecture of the simulation in an effective and easy-to-use manner.

In order to integrate the new functions, two major changes had to be made to the simulation:

1. The underlying data model and database needed to be modified to include parameterized distribution functions, rather than empirical data.
2. Input screens needed to be designed to allow for uploading and editing of these call traffic statistics once they became functioning components of the simulation.

Integration of the theoretical call traffic distributions with the data model allows for the parameterized distributions to be stored and accessed by the simulation. In addition, this directly results in a faster running simulation as discussed in Section 6.2, since instead of

examining potentially hundreds of data points for each empirical distribution, the appropriate traffic statistic could simply be calculated using a closed form solution to the cumulative density function of the given distribution. Since each of the component densities used were exponential, these cumulative density functions existed in closed form, and were easy to calculate and integrate into the simulation engine:

$$P(X \leq x) = \sum_{j=1}^{K} \pi_j \left(1 - e^{-\lambda_j x}\right)$$

Simulation of this random variable simply consists of generating two uniform (0,1) random variables. The first variable is used to choose the component density (by comparison with the mixing parameters), while the second is used to draw from the selected exponential cumulative density function. A copy of the data model can be found in Appendix 5.

Once the parameter data are able to be stored and the random numbers are generated, the only step left is to allow the user some interaction with the parameters. After all, the simulation application is used for a wide range of analysis, and many of these scenarios will require changes to the underlying description of the call traffic. Users of the simulation application include traffic-engineering specialists at TELUS who develop network routing plans, as well as network engineers who plan long-term capacity changes. In order to facilitate user interaction with the parameters, input screens like those found in Appendix 5 were built. Through this screen, the user can get a better intuitive feel for what the component densities represent, and how to make changes to them. In addition, by looking at the component densities of switch pairs which the operator is familiar with, he or she can better understand the various interactions between the multiple parameters. For

instance, the Edmonton network has a single large internet service provider (ISP) which dwarfs all of the other ISPs in the area. All switch pairs originating from other switches and terminating at this ISPs home switch have an abnormally low percentage of type 1 mean calls (short duration). Not only that, but the type 2 mean calls (long duration) have a much higher mean than for any other switch pairs. This finding is consistent with intuition, since one would expect many incoming calls to this switch to be data internet calls which would last much longer. By providing the user with access to information about the parameters, and how they affect different geographic areas of the network, we make the somewhat complicated process of working with a mixture density a little bit easier to understand.

8. Conclusions

Through the COE/TELUS project we have explored new ways of modeling complex random variables using inputs from extremely large data sets. The results of this work will allow TELUS, and potentially other telecommunications companies to reduce overall costs associated with dimensioning a circuit-switched network.

Throughout this thesis, and this project in general, I have shown that mixtures of exponential distributions are a theoretically sound manner of approaching call-traffic modeling. Besides the obvious benefits of calculation time and usability, the underlying value of using these types of distributions is simply the notion that different types of call traffic can and should follow similar distribution patterns with different means.

Recall that I specified two distinct evaluation criteria would be used, goodness-of-fit tests, and validation in the simulation. In addition, I presented two separate data sources to test the chosen methods, call arrival and call duration data.

Given these criteria, mixed exponential distributions appear to be effective for modeling call duration times. Specifically, a mixture of two components densities was adequate when dealing with either residential or business switches; however, moving to a mixture of three component densities only improved performance when dealing with business switches.

Call arrival patterns appear to be modeled well using a single exponential distribution to the interarrival times, thus validating the use of a Poisson arrival process. Further attempts at fitting mixed distributions to these interarrival patterns added little.

The work herein has been presented with respect to its position in the research project as a whole, and more specifically with respect to the pre-existing circuit-switched network simulator. However, it is important to realize that the observations made are applicable to call traffic in general. Hence, the results of this study are equally applicable to other research using call traffic as some form of input. This could include further simulation studies, using similar or different network architectures, for example the newer protocol packet-switching could be analyzed using a simulation with these same call traffic distributions. In addition, analytical work becomes much easier when using a theoretical distribution. The new type of density can be easily understood a mixture of shorter and longer call durations. It is not difficult to explain the significance of each of the parameters, and incorporate these parameters into other types of studies. We have already

mentioned applications like ad-hoc impact analysis of changing overall mixing parameters. In addition, one could introduce whole new components to the system based upon some assumption that a new type of call traffic is emerging.

Call traffic modeling is important in order to design models to keep the network running smoothly. The traditional use of aggregate call volume data can be improved by breaking the data into its constituent components. Further extensions could look at using the component parameters as inputs to other models, such as the optimization techniques developed by other members of the research team. It is certainly possible that using multiple component means to generate scenarios, rather than a series of average cases could improve upon the stochastic nature of the problem which is so difficult to capture within an optimization framework.

Finally, with regards to TELUS, one of the organizations sponsoring this work, new methods of call traffic modeling are becoming more and more important. The changing nature of the telecommunications industry is such that new traffic "types" are emerging all the time. As a result, an analytical method that is explicitly tied to this notion of different characteristics for different types of traffic will certainly be useful in many applications which we cannot anticipate at this time.

Appendix 1

SAS Output – Single Exponential Interarrivals (Business Switches)

---------- Type=B ----------

The CAPABILITY Procedure
Fitted Exponential Distribution for int

Parameters for Exponential Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	0.891718
Mean		0.891718
Std Dev		0.891718

Goodness-of-Fit Tests for Exponential Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.106084		Pr > D	<0.001
Cramer-von Mises	W-Sq	3.962452		Pr > W-Sq	<0.001
Anderson-Darling	A-Sq	37.402607		Pr > A-Sq	<0.001
Chi-Square	Chi-Sq	165.002923	23	Pr > Chi-Sq	<0.001

Quantiles for Exponential Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	0.10000	0.00896
5.0	0.10000	0.04574
10.0	0.10000	0.09395
25.0	0.30000	0.25653
50.0	0.60000	0.61809
75.0	1.20000	1.23618
90.0	2.00000	2.05326
95.0	2.60000	2.67135
99.0	4.00000	4.10651

SAS Output – Single Exponential Interarrivals (Residential Switches)

---------- Type=R ----------

The CAPABILITY Procedure
Fitted Exponential Distribution for int

Parameters for Exponential Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	1.830534
Mean		1.830534
Std Dev		1.830534

Goodness-of-Fit Tests for Exponential Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.0541368		Pr > D	<0.001
Cramer-von Mises	W-Sq	0.7091690		Pr > W-Sq	<0.001
Anderson-Darling	A-Sq	6.9768412		Pr > A-Sq	<0.001
Chi-Square	Chi-Sq	25.3144835	15	Pr > Chi-Sq	0.046

Quantiles for Exponential Distribution

	------Quantile------	
Percent	Observed	Estimated
1.0	0.10000	0.01840
5.0	0.20000	0.09389
10.0	0.20000	0.19287
25.0	0.60000	0.52661
50.0	1.30000	1.26883
75.0	2.50000	2.53766
90.0	4.10000	4.21496
95.0	5.50000	5.48379
99.0	8.20000	8.42992

SAS Output – Single Exponential Durations (Business Switches)

---------- Type=B ----------

The CAPABILITY Procedure
Fitted Exponential Distribution for DURATION

Parameters for Exponential Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	91.09689
Mean		91.09689
Std Dev		91.09689

Goodness-of-Fit Tests for Exponential Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.333		Pr > D	<0.001
Cramer-von Mises	W-Sq	1768.703		Pr > W-Sq	<0.001
Anderson-Darling	A-Sq	24718.687		Pr > A-Sq	<0.001
Chi-Square	Chi-Sq	455652.452	48	Pr > Chi-Sq	<0.001

Quantiles for Exponential Distribution

	--------Quantile--------	
Percent	Observed	Estimated
1.0	0.20000	0.91555
5.0	0.20000	4.67266
10.0	0.30000	9.59801
25.0	0.30000	26.20694
50.0	28.70000	63.14355
75.0	75.60000	126.28710
90.0	193.55000	209.75833
95.0	341.45000	272.90188
99.0	1071.30000	419.51666

SAS Output – Single Exponential Durations (Residential Switches)

---------- Type=R ----------

The CAPABILITY Procedure
Fitted Exponential Distribution for DURATION

Parameters for Exponential Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	162.1915
Mean		162.1915
Std Dev		162.1915

Goodness-of-Fit Tests for Exponential Distribution

Test	----Statistic-----		DF	------p Value------	
Kolmogorov-Smirnov	D	0.30		Pr > D	<0.001
Cramer-von Mises	W-Sq	1341.82		Pr > W-Sq	<0.001
Anderson-Darling	A-Sq	7488.83		Pr > A-Sq	<0.001
Chi-Square	Chi-Sq	3106623.90	44	Pr > Chi-Sq	<0.001

Quantiles for Exponential Distribution

	--------Quantile--------	
Percent	Observed	Estimated
1.0	0.20000	1.63008
5.0	0.20000	8.31934
10.0	5.00000	17.08858
25.0	23.50000	46.65959
50.0	42.40000	112.42258
75.0	113.40000	224.84516
90.0	356.90000	373.45973
95.0	701.60000	485.88230
99.0	2081.00000	746.91945

Appendix 2

SAS Programming Code – NLP Procedure

```
/* Procedure for generating a table of mixed exponential parameters from empirical data
Note that I am using the library "HD" throughout, all relevant files are on the hard
drive of Iridium for reference*/

* STEP 1: Compute MLE for Mixture of 2 Exponentials;

proc nlp
     data=HD.TH_Inters tech=NEWRAP outest=HD.TH_MLEvars_Inter2 maxiter=1000 maxfunc=1000;
     by Type;
     max loglik;
     parms prob=.2, lambda1 = .001, lambda2 = .01;
     bounds lambda1 >= 0, lambda2 >= 0, prob <= 1, prob >0;
     loglik = log(prob*((lambda1)*exp(-(lambda1)*(Int)))+
              (1-prob)*((lambda2)*exp(- (lambda2)*(Int))));
run;

* STEP 2: Extract the relevant parameters from the output data set;

proc SQL;
     CREATE table HD.TH_MLEParms_Inter2 AS
     SELECT Type, prob, lambda1, lambda2
     FROM HD.TH_Mlevars_Inter2
     WHERE _TYPE_ EQ 'PARMS';
run;
```

Appendix 3

C++ Programming Code – EM Algorithm

```
#include <iostream>
#include <math.h>
#include <string>
#include <stdlib.h>
#include <strstream>
#include <fstream>

using namespace std;

#define NumNodes 1
#define NumHours 24
#define MaxRecords 2000
#define RecordsFile "Z:/Thesis/CProgramming/EM/Count24Hr.txt"
#define DataFile "Z:/Thesis/CProgramming/EM/Data24Hr.txt"
#define DetailFile "Z:/Thesis/CProgramming/EM/Detail.txt"
#define SummaryFile "Z:/Thesis/CProgramming/EM/Summary.txt"


int NR[NumNodes][NumNodes][NumHours];                    // Num Records for O/D/Hr
float X[NumNodes][NumNodes][NumHours][MaxRecords];       // Duration data
int O, D, Hr, v;                                         // Current iteration variables

void ReadInData ()
{
        cout << "Reading in record counts...";
        ifstream Count_stream;
        Count_stream.open(RecordsFile, ios::in);
        if (!Count_stream)
        {
                cout << "Error!/nFailed to open record count input file!/n";
                return;
        }
        // used when reading in data from file
        string temp;
        int curO (0), curD (0), curHr (0), prevO (-1), prevD (-1), prevHr (-1);

        // get Column Headings
        getline(Count_stream, temp);

        while(!Count_stream.eof()) // end of file
        {
                getline(Count_stream, temp, ',');
                curO = atoi(temp.c_str());
                getline(Count_stream, temp, ',');
                curD = atoi(temp.c_str());
                getline(Count_stream, temp, ',');
                curHr = atoi(temp.c_str());
                getline(Count_stream, temp);
                NR[curO][curD][curHr] = atoi(temp.c_str());
        }

        Count_stream.close();
        cout << "Completed reading in record counts\n";
        cout << "Max records = " <<MaxRecords <<"\n";

        // Create Data Array

        cout << "Reading in data...\n";
        ifstream data_stream;
        data_stream.open(DataFile, ios::in);
        if (!data_stream)
        {
                cout << "Error!\nFailed to open data input file!\n";
                return;
        }
        // used when reading in data from file
        int curRec (0);

        // get Column Headings
        getline(data_stream, temp);

        while(!data_stream.eof()) // end of file
        {
                getline(data_stream, temp, ',');
                curO = atoi(temp.c_str());
```

```
                getline(data_stream, temp, ',');
                curD = atoi(temp.c_str());
                getline(data_stream, temp, ',');
                curHr = atoi(temp.c_str());
                getline(data_stream, temp);

                if((curO!=prevO) || (curD!=prevD) || (curHr!=prevHr))
                {
                        curRec=0;
                }
                else
                {
                        curRec++;
                }

                X[curO][curD][curHr][curRec] = atof(temp.c_str());
                prevO = curO;
                prevD = curD;
                prevHr = curHr;
        }

        data_stream.close();
        cout << "Completed reading in data\n";
        cout << "Final record = " <<X[NumNodes-1][NumNodes-1][NumHours-1][NR[NumNodes-1][NumNodes-
        1][NumHours-1]-1] <<"\n";

        return;
}

float ExpDensity (float lambda, float x)
{
        return (1/lambda) * (exp(-x/lambda));
}

float MixDensity (float lambda1, float lambda2, float pi1, float pi2, float x)
{
        return ((pi1 * ExpDensity(lambda1,x)) + (pi2 * ExpDensity(lambda2,x)));
}
float CalculateLambda (int j, float lambda1, float lambda2, float pi1, float pi2)
{
        float numerator (0), denominator (0), xi (0);
        int i;

        if(j==1)
        {
                for(i=0; i < NR[O][D][Hr]; i++)
                {
                        xi = X[O][D][Hr][i];
                        numerator += ((ExpDensity(lambda1, xi) * xi) / MixDensity(lambda1, lambda2,
                        pi1, pi2, xi));
                        denominator += (ExpDensity(lambda1, xi) / MixDensity(lambda1, lambda2, pi1,
                        pi2, xi));
                }

        }
        if(j==2)
        {
                for(i=0; i < NR[O][D][Hr]; i++)
                {
                        xi = X[O][D][Hr][i];
                        numerator += ((ExpDensity(lambda2, xi) * xi) / MixDensity(lambda1, lambda2,
                        pi1, pi2, xi));
                        denominator += (ExpDensity(lambda2, xi) / MixDensity(lambda1, lambda2, pi1,
                        pi2, xi));
                }
        }

        return numerator / denominator;
}

float Calculatepi (float lambda1, float lambda2, float pi1, float pi2)
{
        float p (0), xi (0);
        int i;

        for(i=0; i < NR[O][D][Hr]; i++)
        {
                xi = X[O][D][Hr][i];
                p = p + ((( ExpDensity(lambda1, xi) * pi1 )/(MixDensity(lambda1, lambda2, pi1, pi2,
                xi))) / NR[O][D][Hr]);
        }
        return p;
}

void EMAlgorithm ()
{
        bool stop (0);
        float lambda1, lambda2, pi1, pi2, newLambda1, newLambda2, newpi1, newpi2, tl1, tl2, tp1;
        ofstream Detail_stream, Summary_stream;
        Detail_stream.open(DetailFile, ios::out);
```

```
        if (!Detail_stream)
        {
                cout << "Error!/nFailed to open output file!/n";
                return;
        }
        Summary_stream.open(SummaryFile, ios::out);
        if (!Summary_stream)
        {
                cout << "Error!/nFailed to open output file!/n";
                return;
        }
        int i,j,k;
        for(i = 0; i < NumNodes; i++)
        {
                for(j = 0; j < NumNodes; j++)
                {
                        for(k = 0; k < NumHours; k++)
                        {
                        O = i;
                        D = j;
                        Hr = k;
                        lambda1 = 1000;
                        newLambda1 = 1000;
                        lambda2 = 10;
                        newLambda2 = 10;
                        pi1 = .8;
                        newpi1 = .8;
                        pi2 = .2;
                        newpi2 = .2;
                        v=0;
                        while(!stop)
                        {
                        // Output existing parameters & iteration info
                        Detail_stream <<O <<"," <<D <<"," <<Hr <<"," <<v <<"," <<lambda1 <<","
                        <<lambda2 <<"," <<pi1 <<"," <<pi2 <<"\n";
                        cout <<O <<"," <<D <<"," <<Hr <<"," <<v <<"," <<lambda1 <<"," <<lambda2 <<","
                        <<pi1 <<"," <<pi2 <<","<<tl1 <<"\n";

                        //Calculate new parameters (v+1)
                        newLambda1 = CalculateLambda (1, lambda1, lambda2, pi1, pi2);
                        newLambda2 = CalculateLambda (2, lambda1, lambda2, pi1, pi2);
                        newpi1 = Calculatepi (lambda1, lambda2, pi1, pi2);
                        newpi2 = (1 - newpi1);

                        //Check for stopping criteria
                        tl1 = lambda1 - newLambda1;
                        tl2 = lambda2 - newLambda2;
                        tp1 = pi1 - newpi1;

                        if((lambda1-newLambda1 < (0.0001)) && (lambda1-newLambda1 > (-0.0001))
                        && (lambda2-newLambda2 < (0.0001)) && (lambda2-newLambda2 > (-0.0001))
                                && (pi1-newpi1 < (0.0001)) && (pi1-newpi1 > (-0.0001)) || v==999)
                          {
                                stop = 1;
                        }
                        lambda1 = newLambda1;
                        lambda2 = newLambda2;
                        pi1 = newpi1;
                        pi2 = newpi2;
                        v++;
                }
                Summary_stream <<O <<"," <<D <<"," <<Hr <<"," <<v <<"," <<lambda1 <<"," <<lambda2
                <<"," <<pi1 <<"," <<pi2 <<"," <<tl1 <<"," <<tl2 <<"," <<tp1 <<"\n";
                stop = 0;
                }
                }
        }
}

int main ()
{
        ReadInData ();
        EMAlgorithm ();
        return 0;
}
```

Appendix 4

Detailed Convergence Results of NLP Method

TYPE: B

Optimization Start

Active Constraints			0	Objective Function				-310653.565
Max Abs Gradient Element			16965091.246					

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1*	0	4	0	-304669	5984.6	17494736	0.0609	-99872
2*	0	5	0	-296611	8058.4	13930829	0.244	-36426
3*	0	6	0	-274348	22262.1	6210408	0.975	-32251
4*	0	7	0	-260281	14067.1	2339511	1.000	-20680
5*	0	8	0	-254850	5430.9	615690	1.000	-8561.8
6*	0	9	0	-254015	835.7	71235.3	1.000	-1460.8
7*	0	14	0	-252711	1303.4	709379	3.409	-251.4
8*	0	15	0	-241386	11325.2	47757.2	1.000	-16161
9*	0	16	0	-231558	9827.8	49342.6	1.000	-14755
10*	0	17	0	-225227	6331.4	6416.5	1.000	-9570.4
11*	0	18	0	-221831	3395.6	2361.6	1.000	-5328.1
12*	0	19	0	-220786	1045.3	609.1	1.000	-1759.8
13*	0	20	0	-220674	111.8	69.0146	1.000	-206.6
14*	0	21	0	-220673	1.6797	1.5424	1.000	-3.301
15*	0	22	0	-220673	0.000855	0.00718	1.000	-0.0017

Optimization Results

Iterations	15	Function Calls	23
Hessian Calls	16	Active Constraints	0
Objective Function	-220672.6812	Max Abs Gradient Element	0.0071835669
Slope of Search Direction	-0.00170308	Ridge	6.0540242954

GCONV convergence criterion satisfied.

TYPE: R

Optimization Start

Active Constraints			0	Objective Function				-238210.1138
Max Abs Gradient Element			12324741.001					

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1*	0	5	0	-236336	1873.8	11956816	0.0387	-48916
2*	0	6	0	-232577	3759.2	10317430	0.155	-25739
3*	0	7	0	-221862	10715.5	5832576	0.619	-21868
4*	0	8	0	-213575	8287.0	1913179	1.000	-12175
5*	0	9	0	-210746	2828.2	310065	1.000	-4157.3
6*	0	10	0	-210098	648.1	13615.4	1.000	-1000.8
7*	0	11	0	-210071	27.5796	793.5	1.000	-49.008
8*	0	12	0	-210070	0.2612	16.0056	1.000	-0.486
9*	0	13	0	-210070	0.00137	0.6511	1.000	-0.0026
10*	0	14	0	-210070	2.262E-6	0.0137	1.000	-442E-8

Optimization Results

Iterations	10	Function Calls	15
Hessian Calls	11	Active Constraints	0
Objective Function	-210070.4461	Max Abs Gradient Element	0.0136823908
Slope of Search Direction	-4.420914E-6	Ridge	2021.2081392

GCONV convergence criterion satisfied.

Detailed Convergence Results of EM Algorithm Method

Business Switches

Iteration	Lambda1	Lambda2	p1	p2
1	40.8526	226.0310	0.7288	0.2712
2	30.8567	287.8670	0.7676	0.2324
3	28.0481	312.1790	0.7782	0.2218
4	27.1351	314.2710	0.7773	0.2227
5	26.6709	311.1270	0.7734	0.2266
6	26.3076	307.1880	0.7693	0.2307
7	25.9985	303.4700	0.7655	0.2345
8	25.7098	300.1150	0.7618	0.2382
9	25.4537	297.0070	0.7581	0.2419
10	25.1991	294.0820	0.7548	0.2452
11	24.9664	291.2900	0.7517	0.2483
12	24.7517	288.7720	0.7488	0.2512
13	24.5515	286.5040	0.7461	0.2539
14	24.3627	284.4250	0.7435	0.2565
15	24.1854	282.4800	0.7411	0.2589
16	24.0180	280.6730	0.7385	0.2615
17	23.8539	278.8860	0.7362	0.2638
18	23.6969	277.2310	0.7340	0.2660
19	23.5491	275.6630	0.7320	0.2680
20	23.4049	274.1780	0.7300	0.2700
21	23.2659	272.7930	0.7282	0.2718
22	23.1352	271.4530	0.7264	0.2736
23	23.0181	270.1990	0.7246	0.2754
24	22.9011	269.0140	0.7229	0.2771
25	22.7835	267.7680	0.7214	0.2786
26	22.6784	266.6570	0.7195	0.2805
27	22.5671	265.5100	0.7179	0.2821
28	22.4559	264.4580	0.7163	0.2837
29	22.3483	263.4190	0.7148	0.2852
30	22.2488	262.4100	0.7133	0.2867
31	22.1529	261.4360	0.7119	0.2881
32	22.0553	260.5430	0.7105	0.2895
33	21.9690	259.6540	0.7092	0.2908
34	21.8780	258.8270	0.7080	0.2920
35	21.7963	258.0140	0.7067	0.2933
36	21.7105	257.2490	0.7056	0.2944
37	21.6346	256.5030	0.7043	0.2957
38	21.5528	255.7600	0.7032	0.2968
39	21.4796	255.0460	0.7020	0.2980
40	21.4012	254.3550	0.7009	0.2991
41	21.3312	253.6740	0.6999	0.3001
42	21.2564	253.0180	0.6986	0.3014
43	21.1853	252.3190	0.6974	0.3026
44	21.1092	251.6290	0.6963	0.3037
45	21.0299	250.9540	0.6952	0.3048
46	20.9584	250.2720	0.6940	0.3060
47	20.8810	249.6340	0.6929	0.3071
48	20.8102	248.9730	0.6918	0.3082
49	20.7344	248.3380	0.6908	0.3092
50	20.6658	247.7010	0.6897	0.3103
51	20.5909	247.0920	0.6887	0.3113
52	20.5237	246.4790	0.6876	0.3124
53	20.4509	245.8550	0.6866	0.3134
54	20.3848	245.2510	0.6856	0.3144
55	20.3125	244.6780	0.6846	0.3154
56	20.2487	244.0700	0.6835	0.3165
57	20.1868	243.5020	0.6825	0.3175
58	20.1184	242.9520	0.6815	0.3185
59	20.0536	242.3680	0.6804	0.3196
60	19.9840	241.8140	0.6794	0.3206
61	19.9185	241.2270	0.6784	0.3216
62	19.8479	240.6700	0.6772	0.3229
63	19.7709	240.0170	0.6760	0.3240
64	19.6993	239.3860	0.6748	0.3252
65	19.6205	238.7660	0.6737	0.3263
66	19.5462	238.1260	0.6725	0.3275
67	19.4661	237.4990	0.6713	0.3287
68	19.3917	236.8370	0.6701	0.3299
69	19.3115	236.1720	0.6689	0.3311
70	19.2265	235.5470	0.6676	0.3324
71	19.1461	234.9090	0.6664	0.3336
72	19.0608	234.1100	0.6651	0.3349
73	18.9713	233.4170	0.6638	0.3362
74	18.8869	232.6700	0.6624	0.3376
75	18.7949	231.9350	0.6611	0.3389
76	18.6997	231.2000	0.6596	0.3404
77	18.6076	230.5900	0.6580	0.3420
78	18.5119	229.8330	0.6566	0.3434
79	18.4100	229.0030	0.6550	0.3450
80	18.3038	227.8340	0.6532	0.3468
81	18.1869	227.6490	0.6513	0.3487

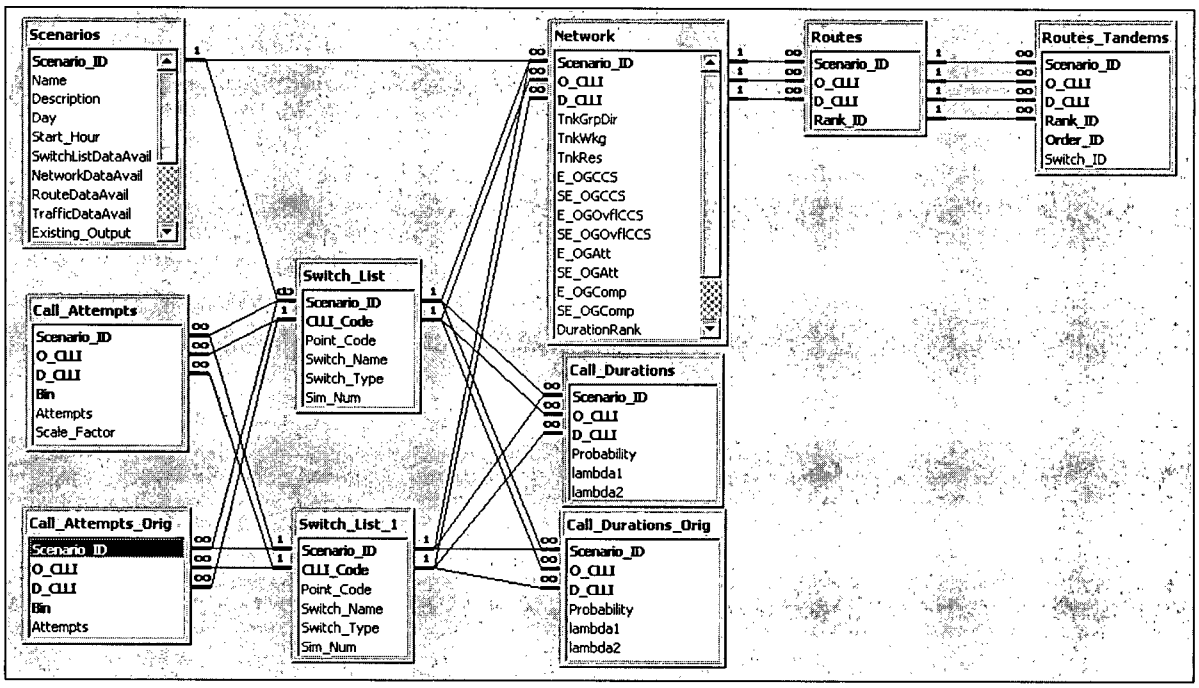
Residential Switches

Iteration	Lambda1	Lambda2	p1	p2
1	55.0882	380.5520	0.6710	0.3290
2	46.6506	466.0670	0.7245	0.2755
3	45.4593	514.3350	0.7510	0.2490
4	46.0070	541.2640	0.7654	0.2346
5	46.8025	559.2720	0.7748	0.2252
6	47.5195	572.8110	0.7817	0.2183
7	48.1068	583.3670	0.7868	0.2132
8	48.5713	591.6320	0.7908	0.2092
9	48.9323	598.1910	0.7938	0.2062
10	49.2134	603.2440	0.7961	0.2039
11	49.4344	607.1560	0.7978	0.2022
12	49.6047	610.2600	0.7991	0.2009
13	49.7348	612.6110	0.8002	0.1998
14	49.8381	614.4770	0.8010	0.1990
15	49.9175	615.9200	0.8017	0.1983
16	49.9801	617.0380	0.8021	0.1979
17	50.0254	617.8820	0.8025	0.1975
18	50.0645	618.5520	0.8028	0.1972
19	50.0934	619.0850	0.8030	0.1970
20	50.1148	619.4700	0.8032	0.1968
21	50.1304	619.7640	0.8033	0.1967
22	50.1441	620.0000	0.8034	0.1966
23	50.1538	620.1950	0.8035	0.1965
24	50.1617	620.3290	0.8035	0.1965
25	50.1673	620.4400	0.8036	0.1964
26	50.1718	620.5170	0.8036	0.1964
27	50.1756	620.5780	0.8037	0.1964
28	50.1783	620.6140	0.8037	0.1963
29	50.1802	620.6510	0.8037	0.1963
30	50.1816	620.6830	0.8037	0.1963
31	50.1827	620.7130	0.8037	0.1963
32	50.1836	620.7310	0.8037	0.1963
33	50.1843	620.7450	0.8037	0.1963
34	50.1848	620.7570	0.8037	0.1963
35	50.1852	620.7670	0.8037	0.1963
36	50.1855	620.7770	0.8037	0.1963
37	50.1857	620.7600	0.8037	0.1963
38	50.1858	620.7600	0.8037	0.1963
39	50.1858	620.7600	0.8037	0.1963
40	50.1858	620.7610	0.8037	0.1963

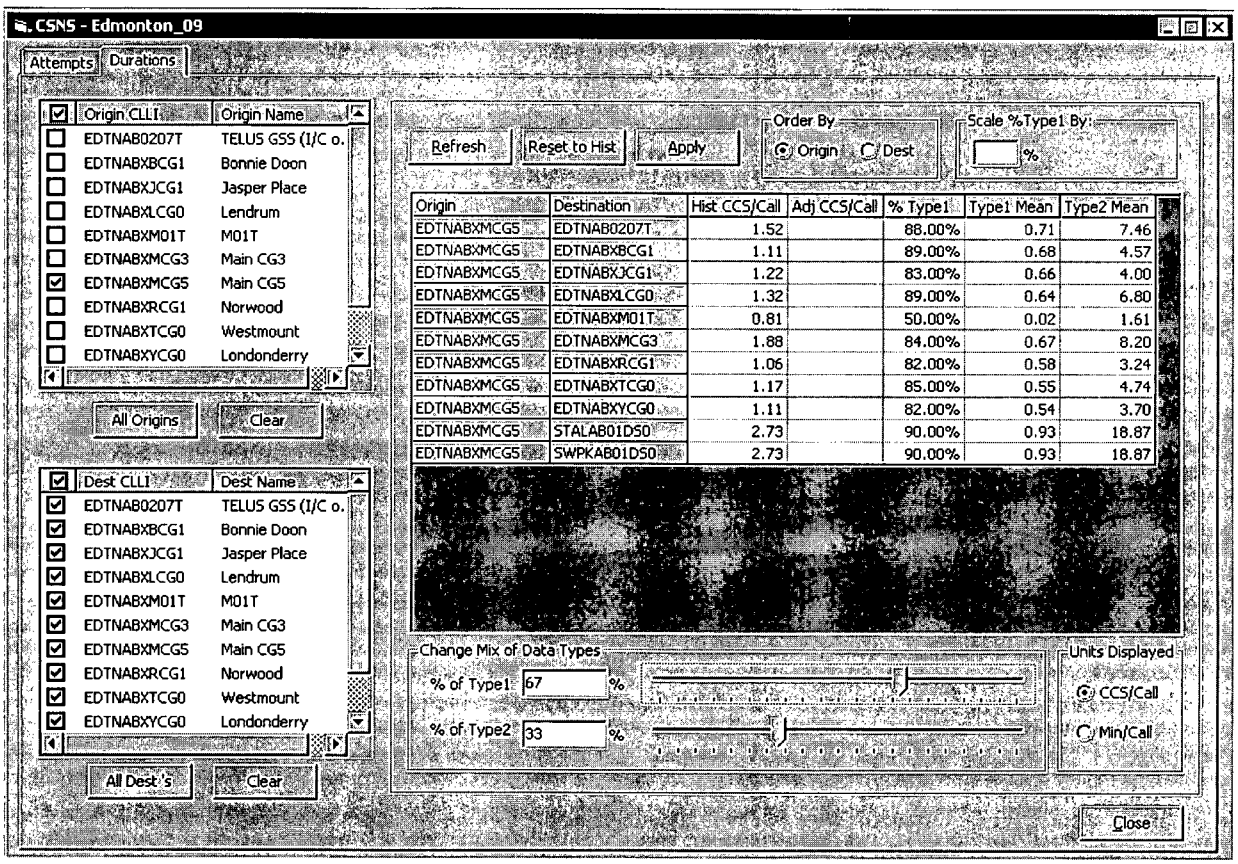
82	18.0722	226.9860	0.6496	0.3504
83	17.9562	226.8350	0.6478	0.3522
84	17.8444	226.3320	0.6461	0.3539
85	17.7318	222.7690	0.6444	0.3556
86	17.5908	221.3450	0.6418	0.3582
87	17.4272	220.2120	0.6392	0.3608
88	17.2603	218.9580	0.6365	0.3635
89	17.0777	217.5910	0.6336	0.3664
90	16.8888	216.1890	0.6303	0.3697
91	16.6828	214.7570	0.6270	0.3730
92	16.4613	214.4040	0.6235	0.3765
93	16.2399	210.7460	0.6200	0.3800
94	15.9764	208.8670	0.6156	0.3844
95	15.6851	206.9900	0.6109	0.3891
96	15.3727	204.9620	0.6058	0.3942
97	15.0324	202.6230	0.6000	0.4000
98	14.6555	200.9010	0.5938	0.4062
99	14.2424	196.9760	0.5871	0.4129
100	13.7613	194.1160	0.5791	0.4209
101	13.2179	191.1280	0.5700	0.4300
102	12.5976	187.7500	0.5598	0.4402
103	11.8790	183.9830	0.5479	0.4521
104	11.0410	179.8600	0.5341	0.4660
105	10.0470	175.1120	0.5176	0.4824
106	8.8486	169.8000	0.4974	0.5026
107	7.3835	162.8750	0.4730	0.5270
108	5.5856	155.7520	0.4423	0.5577
109	3.4558	147.8070	0.4052	0.5948
110	1.3481	139.7330	0.3659	0.6341
111	0.3821	135.1420	0.3447	0.6553
112	0.2976	133.9310	0.3412	0.6588
113	0.2936	133.7700	0.3404	0.6596
114	0.2935	133.8290	0.3403	0.6597
115	0.2935	133.8200	0.3403	0.6597
116	0.2934	133.8200	0.3403	0.6597

Appendix 5

Screenshot of Integrated Data Model



Screenshots of modified simulation application



References

Agha, M. and Ibrahim, M.T. "Algorithm AS 203: Maximum Likelihood Estimation of Mixtures of Distributions." *Applied Statistics*, Vol. 33, No. 3. (1984), pp. 327-332.

Bratley, P., Fox, B. and Schrage, L. *A Guide To Simulation*. New York: Springer-Verlag. 1983.

Braun, D. *Efficient Routing of Telephone Calls in a Circuit-switched Network*. MSc Thesis, unpublished. University of British Columbia (1999).

Conover, W. *Practical Non-parametric Statistics, 3rd ed.* New York: Wiley. 1999.

Day, N.E. "Estimating the Components of a Mixture of Normal Distributions." *Biometrika,* Vol. 56 (1969), pp 463-474.

Dempster, A.P., Laird, N.M., and Rubin, D.B. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1. (1977), pp. 1-38.

Hasselblad, V. "Estimation of Finite Mixtures of Distributions from the Exponential Family." *Journal of the American Statistical Association*, Vol. 64, No. 328. (Dec., 1969), pp. 1459-1471.

Kao, E. *An Introduction to Stochastic Processes*. Belmont, CA: Wadsworth. 1997.

Kabanuk, S. *Nonhierarchical Alternate Routing in A circuit-switched network at TELUS.* MSc Thesis, unpublished. University of British Columbia (2000).

Lindsay, J.K. and Roeder, L. "Residual Diagnostics for Mixture Models." *Journal of the American Statistical Association*, Vol. 87, No. 419. (Sep., 1992), pp. 785-794.

McLachlan, G.J. "On the Choice of Starting Values for the EM Algorithm in Fitting Mixture Models." *Statistician*, Vol. 37, No. 4/5. (1988), pp. 417-425.

Murray, G.D., and Titterington, D.M. "Estimation Problems with Data from a Mixture." *Applied Statistics*, Vol. 27, No. 3. (1978), pp. 325-334.

Redner, R.A. and Walker, H.F. "Mixture Densities, Maximum Likelihood and the EM Algorithm." *SIAM Review*, Vol. 26, No. 2. (Apr., 1984), pp. 195-239.

Sim, T. *Determining Trunk Reservation Parameters In a Non-Symmetric Telecommunications Network With Non-Uniform Demand. MSc Thesis*, unpublished. University of British Columbia (2001).

Smith, I. *An application of Multivariate Analysis to Time of Day Routing in Telecommunications Networks*. MSc Thesis, unpublished. University of British Columbia (2000).

Wang, P.M. and Puterman, M.L. "Mixed Logistic Regression Models". *Journal of Agricultural, Biological, and Environmental Statistics,* Vol.3 No.2 (1998), pp175-200.

Wolfe, J.H. "Pattern Clustering by Multivariate Mixture Analysis." *Multivariate Behavioral Research*, Vol.5 (1970), pp. 329-350.

Wu, C.F.J. "On the Convergence Properties of the EM Algorithm." *Annals of Statistics,* Vol. 11, No. 1. (Mar., 1983), pp. 95-103.