NONHIERARCHICAL ALTERNATE ROUTING IN A CIRCUIT-SWITCHED NETWORK AT TELUS

by

Steven Joseph Kabanuk

Bachelor of Commerce, University of Alberta, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN BUSINESS

in

THE FACULTY OF GRADUATE STUDIES

(Department of Commerce and Business Administration)

We accept this thesis as conforming
To the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 2000

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_

The University of British Columbia
Vancouver, Canada

Date _December 18, 2000_

ABSTRACT

The primary objective of this project was to identify opportunities and develop solutions that will enable TELUS to reduce the cost of maintaining and improving a portion of their existing local network in British Columbia, Canada. To realize this objective, the focus was on optimal routing of calls assuming a fixed network capacity. Our hypothesis was that by changing the routing rules between an origin switch and a destination switch, we could alleviate pressure from congested areas in the network by diverting traffic through less utilized areas.

Three types of optimization models have been developed: two linear programs, a mixed-integer model, and a nonlinear program. Irrespective of their objective functions, the purpose of each model was to suggest new sets of routes for each pair of switches. Development of these models and assessment of their ability to suggest routes which reduced the number of blocked calls in the simulated network was the central focus of this thesis.

The modeling results were compared with the existing routing rules in a network simulator. It appears that TELUS could benefit from implementation of alternate nonhierarchical routing rules. Specifically, employing the combined routing rules – that attempt the direct route first, then attempt the 2-link Maximum Linear Model routes from the 11:00-12:00 time interval, and finally the currently used alternate routes – generated the best routing table of those tested. Furthermore, these routing rules can be employed throughout the day without the need for change. The results from the trunk reservation analysis suggest that implementation of a uniform policy would be of marginal value.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I.   INTRODUCTION

What Is Traffic Engineering?

Traffic engineering is nearly as old as Alexander Graham Bell's telephone. With the increased demand for telecommunications, more sophisticated infrastructure became increasingly widespread. This growth created opportunities for efficiencies from better traffic engineering practices. Agner Krarup Erlang (1878 - 1929) performed pioneering work in the early 1900s, and could be considered the founding father of traffic engineering. His approximations for the relationship between offered and carried traffic are still widely used to model voice traffic today. From this foundation, traffic engineering has evolved into three areas: routing, dimensioning, and monitoring defined below. Change in any one of these areas affects the other two. Consequently, one must consider the interrelationship of routing, dimensioning, and monitoring in order to study routing; it cannot be studied in isolation.

For a proper introduction to routing, one must first become familiar with necessary network nomenclature. End-users employ communication devices such as telephones, fax machines, and modems that are connected to switches, which in turn are connected to other communication devices and switches. The direct connection between two switches is called a *trunk group (or a link)* and each trunk group is comprised of several *trunks*. A set of communication devices, switches and trunk groups is called a telecommunication *network*. A connection between two communications devices is achieved through visiting at least one switch, and quite possibly a series of intermediate switches, via trunk groups. The source switch, the visited intermediate switches and the destination switch of a transmission in a network is called a *route*. The end-user is unaware that there is a set of possible routes to try in the event that no idle capacity exists on the first choice route. In traffic engineering, *routing* means determining the best set of routes between each pair of switches. The degree of flexibility for routing decisions is dictated by the technology used in the network and the capabilities of the network monitoring systems.

The science of dimensioning arose in response to continued growth for telecommunication services. Historically, telecommunication companies (telcos) were provincial monopolies in Canada so the Canadian Radio-television Telecommunications Commission (CRTC) required a minimum quality of service that telcos were required to maintain. The quality of service was measured by the number of attempted calls that were blocked as a fraction of the total number of calls attempted. Typically, the required grade of service was P.01, which translated to a maximum of 1 call that could be blocked out of every 100 attempted. Post deregulation, traffic engineers still use the same measurement scheme to determine the quality of service, but the ultimate decision is in the hands of the telco rather than the regulator. In order to maintain the service quality, telcos have to continually invest in their networks. *Dimensioning* is the science of determining when and where capacity should be added to the network in order to maintain a desired quality of service.

The final responsibility of traffic engineers is to monitor the network. This is performed for three main reasons: performance tracking, forecasting, and real-time management. Firstly, performance tracking is necessary to monitor the quality of service and to observe the network's response to previous decisions. Secondly, the historical performance of the network is a rich source of information that can aid an engineer's decisions, which in turn will affect future performance. Finally, real-time management is necessary to respond to short-term fluctuations in traffic and network conditions. For example, a radio show could be giving away prizes to listeners if they call in. The result is a large spike in network traffic as people attempt to call. If the engineers allow all these calls to enter the network, they will not be able to maintain an adequate quality of service for other network traffic. Another example could be switch or group failure causing an immediate rerouting of traffic.

Traffic Management Strategies

There are myriad traffic management strategies, and one could expend considerable effort comparing the strengths and weaknesses of each. We will limit the discussion to the two possible traffic management

strategies in the studied network. Given the technological capabilities of the network studied, implementation of these two strategies can occur without the need for capital investment. Specifically, we will discuss *alternate hierarchical routing* (AHR) and *alternate nonhierarchical routing* (ANHR) in a circuit-switched network. Trunk reservation warrants discussion because network instabilities can occur when ANHR is implemented without *trunk reservation*.[1]

Circuit Switching

Historically, the purpose of a telecom network was to provide a connection between different users for the purpose of verbal communication. This connection was achieved by having a caller's telephone connected to the called party's telephone via a series of trunks and switches. The complete connection from the calling party to the called party was called a *circuit*.[2] The circuit utilizes one trunk on each link in the circuit. In other words, no other call could use the trunks in the circuit until the current phone call completed. Although more sophisticated technology is available, circuit switching is still widely used in telcos' networks today.

Alternate Hierarchical Routing (AHR)

Alternate routing is combined with hierarchical routing to create AHR. In an alternate routing system, a call has multiple routes it can attempt to complete a call. If the initial route does not have an available circuit, then the first alternate route is attempted. If the first alternate does not have a free circuit, the second alternate is attempted. This sequence continues until the call finds a route with an available circuit or all prespecified options have been exhausted. In a *loss system*, such as the one used by TELUS and most North American telcos, the user receives a network busy signal to indicate that their call was blocked upon exhaustion of all possible routes. In a hierarchical routing scheme, switches have different roles designated by their *class*. In local networks, engineers use the numbers 4 and 5 to classify switches.

---

[1] See the literature review in this thesis for supporting evidence.
[2] A circuit is the physical connection of a route.

A class 4 switch can serve as *tandem*,[3] whereas a class 5 switch does not. Consequently, only the designated class 4 switches can be intermediary switches on a route.

A hierarchical network is further subdivided into sectors, each having its own tandem. In Figure 1, switch E is the tandem for sector 1 and switch F is the tandem for sector 2. Other than the direct route, calls within a sector only have one alternate route in the network illustrated. For example, if an end-user connected to switch A wants to call a user connected to switch B, the first attempted route will be (A→B) – the *direct* route. The second, and in this case, final route will be (A→E→B). Traffic from A to B that traverses route (A→E→B) is termed *overflow* because it could not be carried on the first route. Calls traversing sectors have three possible routes in the network illustrated. If a user connected to switch B wants to call a user connected to switch C, an alternate hierarchical network would initially try route (B→C), then (B→F→C) and finally (B→E→F→C).
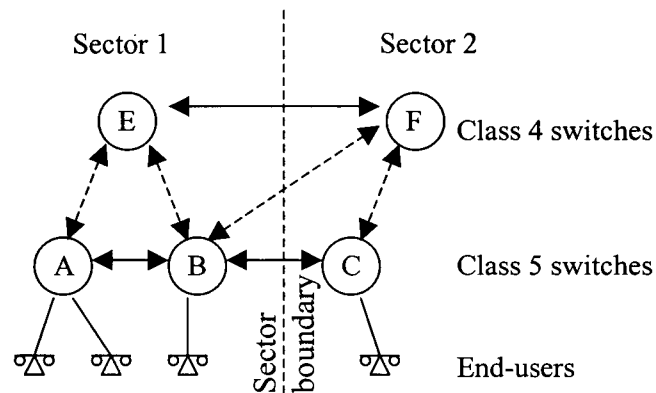
Figure 1 – Alternate hierarchical routing from/to switch B

Alternate Nonhierarchical Routing (ANHR)

In AHNR networks all local switches have class 4 capability and thus are able to serve as tandems. Otherwise, the functionality of ANHR networks is identical to that of AHR networks. The number of intermediate switches is generally restricted to a maximum of 1 or 2 for the following reason: a single call

---

[3] A tandem is an intermediary switch in a circuit (i.e., not the first nor last switch in a route).

will utilize 1 trunk on each link, thus shorter routes are more efficient. For example, the resources needed to connect a call on a 5-link route could theoretically be used to complete 5 direct calls.

Trunk Reservation

There is a positive correlation between network utilization and overflow traffic. As direct connections start to fill up, traffic is forced onto alternate routes. If an alternate route uses two links, then two potential circuits on direct links can no longer be used. This may cause further traffic to overflow onto their 2-link (or longer) alternate routes. More 2-link circuits being used causes more overflow. As this behaviour propagates through the network, the number of new calls that can be completed drops off dramatically. This behaviour is termed the *knock-off effect*. The protective mechanism is to reserve a portion of the circuits in each trunk group for direct calls only; this has the result of stabilizing network performance by inhibiting the knock-off effect.[4]

Problem Description

The primary objective of this project was to identify opportunities and develop solutions that will enable TELUS to reduce the cost of maintaining and improving a portion of their existing local network in British Columbia, Canada. To realize this objective, the focus was on optimalrouting of calls assuming a fixed network capacity. Currently, AHR with circuit-switched technology is implemented in the network studied. By changing the routing rules, ANHR was implemented in a simulated local network situated in British Columbia. Our hypothesis was that by changing the routing rules between an origin switch and a destination switch (i.e., between an *OD-pair*[5]), we could alleviate pressure from congested areas in the network by diverting traffic through less utilized areas. Consequently, the need to add more trunks to the network would diminish or be mitigated altogether.

---

[4] See chapter II for a more detailed discussion of trunk reservation.
[5] As an example of an origin-destination pair, if a call followed the route A→B→C→D, the OD-pair would be A-D. For the route A→F→G→H→D, the OD-pair would still be A-D. Thus in this example, the OD-pair A-D has two different routes associated with it.

Project Overview

Four main modules have been developed to fulfill the primary objective: data analysis, optimization models, multivariate analysis, and network simulation (Figure 2). Each module is briefly discussed below.
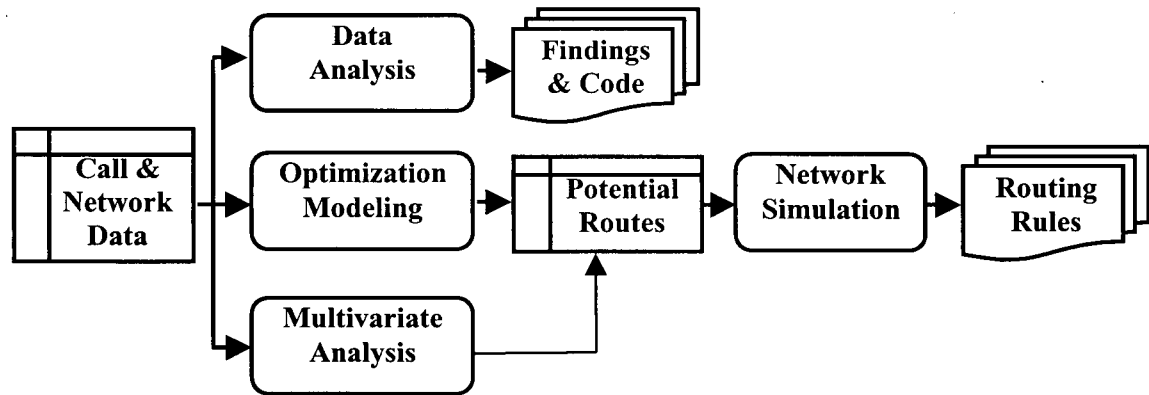
Figure 2 – The project framework

Data Analysis

The purpose was to identify time-series patterns, to identify cross-sectional patterns and to assess the quality of data. The research team [4] has studied call demand patterns over varying time intervals (minutes, hours, days and weeks). Second, they analyzed the movement of call concentration zones throughout the day to investigate whether pursuing time-of-day routing was of value.[6] Furthermore, data analysis has allowed the research team to perform consistency checks on our interpretation of network functionality.

Optimization Models

Three types of optimization models have been developed: two linear programs, a mixed-integer model, and a nonlinear program. Irrespective of their objective functions, *the purpose of each model was to suggest new sets of routes for each OD-pair*. One should note that all models are an abstraction of reality

---

[6] Implementation of *time-of-day routing* means that the routing rules can be changed once or twice at different times of the day.

and simplifying assumptions were necessary to keep them tractable. However, these assumptions may not hold true for the real network; therefore, finding optimal solutions for a particular model does not guarantee that an optimal set of routing rules will be generated for the real network. Hence suggested rules required testing before they are presented to the company for implementation. The modeling results were compared with the existing routing rules in the network simulator (see chapter V). *Development of these models and assessment of their ability to reduce the number of blocked calls in the simulated network was the central focus of this thesis.*

Currently, routing rules in the network are not changed at different times of the day and TELUS was interested in determining whether changing the rules would be a value. Consequently, a second objective was to assess the benefit of implementing time-of-day routing. The models, which attempted to optimize network performance within discrete time intervals, could suggest different sets of rules for different intervals.

Multivariate Analysis

Demand for resources in portions of the network can change by time of day. For example, business regions often experience peak load during the day, whereas suburb-based switches experience peak load at night. Consequently, opportunities exist to reroute traffic in order to utilize idle capacity at different times of the day. Principle component analysis followed by cluster analysis has identified differences in time-of-day demand patterns, suggest time-of-day routing rules and estimate their impact. This research was performed by other members of the research team concurrent to the development of the optimization models [6].

Network Simulation

The simulation is an abstraction of a circuit-switched network that permits assessment of the performance of alternative sets of routing rules. The inputs of the simulation consist of a time interval, distributions for call length and call frequency, link capacities, and routing rules. The simulation generates several

statistics such as number of calls completed, calls blocked, trunk group utilization and the percentage of calls that traversed each alternate route. The simulation was validated by comparing several performance metrics against historical call data and it serves as a litmus test for all recommendations from the optimization models and multivariate analysis.

II.   LITERATURE REVIEW

Stabilization of Alternate Routing Networks

Krupp, Roy S.

IEEE 1982 International Communications Conference, Vol. 1 & 2

The author demonstrated that the performance of an alternate hierarchical routing (ANHR) without trunk reservation has inherent instabilities in symmetric and nonsymmetric networks.[7] The Erlang-B formula[8] was implemented in a system of equations used to represent both a 10-node and a 100-node network for the symmetric case. Results indicate carried load cannot be uniquely mapped from offered traffic. Specifically, it was determined that two equilibrium states for carried traffic exist for most levels of offered traffic. Simulations of the same network provide supporting evidence of the equilibrium state instability. When combined with instability, hysteresis between the two states was observed in some cases, which resulted in a significant decrease in carried load. The performance of both the analytic system and the simulation, when a small number of trunks are reserved, generated a marked increase in carried traffic. The simulation was further used to show the dramatic swings in carried load over time for a system with no trunk reservation. This result was contrasted with a network containing a small number of reserved trunks on each link (4 out of 100 trunks). Reservation of the trunks stabilized carried traffic at

---

[7] In a symmetric network, trunk group sizes (the number of connections between two switches) are identical for all links in the network, whereas nonsymmetric networks have varying trunk group sizes.
[8] A widely used nonlinear function that calculates the carried traffic on a link given the offered traffic and link capacity.

a higher level than the unreserved scenario. Simulation of a 10-node nonsymmetric network provided conclusions consistent with the symmetric case. The results indicate that implementation of trunk reservation yields a higher carried traffic load in network with ANHR.

The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing

Akinpelu, J. M.

AT & T Bell Laboratories Technical Journal, Vol. 63, No. 7, September 1984

With Krupp's work[9] as a foundation, the author extended the mathematical models developed to more general nonhierarchical networks. The developed system of equations is capable of analyzing networks with and without trunk reservation. To support analytical results, scenarios were tested in the simulation developed by Krupp. Multiple values for carried load were observed for the same combination of offered loads, trunk group sizes, routing rules, and reserved trunks in a 10 node symmetric network. The models were applied to a nonsymmetric, nonhierarchical 8-node network with nonuniform offered load. The author found that the percentage of blocked calls with respect to percent overload had multiple values when percent overload was in the 0-4% range.

Hierarchical and nonhierarchical, symmetric networks containing 25, 30, and 140 nodes were studied under uniform and nonuniform loads with and without trunk reservation. The 25-node network was the initial choice for the 1986 deployment of dynamic nonhierarchical routing (DHNR) and the 140-node network was "the 1989 DHNR network envisioned in the predivestiture environment" for Bell. The nonhierarchical networks had an average of seven percent reserved trunks and approximately nine percent fewer trunks than their hierarchical counterparts.

The author concluded that network efficiency is related to the system's ability to favour 1-link calls over multilink calls and that 5% trunk reservation in a nonhierarchical network carries more 1-link calls than the identical unreserved network. A second conclusion was that symmetric nonhierarchical network performance improves as trunk group size increases. Krupp found that performance decreased. However, these two seemingly contradictory findings were based upon analysis of networks of different sizes which does not facilitate a good basis for comparison. Finally, hierarchical routing outperforms nonhierarchical routing in all configurations tested. However, there is no discussion of the specifics of the implemented hierarchy or the engineering of trunk group sizes with respect to offered loads.

III. THE NETWORK – TOPOLOGY AND ACTIVITY

The Network under Study

Takeovers, mergers and continued capital investment all contribute to TELUS having heterogeneous technology dispersed throughout their operations. Each type of technology requires a different management strategy and knowledge base. We are focusing on a portion of TELUS' local voice network in British Columbia, which exclusively employs circuit-switched technology and alternate hierarchical routing (AHR). This network contains 32 large switches in 2 sectors. As mentioned in chapter I, each sector contains one tandem, therefore 2 of the 32 switches are set as class 4 tandem switches. From this point forward, we will refer to this network as *the network.*

Trunks Working and Capacity

A common unit of measure in North America for traffic is the centi-call second (CCS – i.e., 100 call seconds). If one call lasts one hour, it would consume 36 CCS. In other words, one trunk can carry a theoretical maximum traffic load of 36 CCS in an hour. *Trunks working* is a measure of the number of 1-link circuits directly connecting two switches. *Trunk group capacity* is determined by multiplying the

---

[9] See the previous review.

number of trunks working by a unit time. For example, a link with 10 trunks working could carry a theoretical maximum of 360 CCS in an hour. The capacity in CCS of each link in the network is displayed in Figure 3.

Link Capacity for i-j Switch Pairs
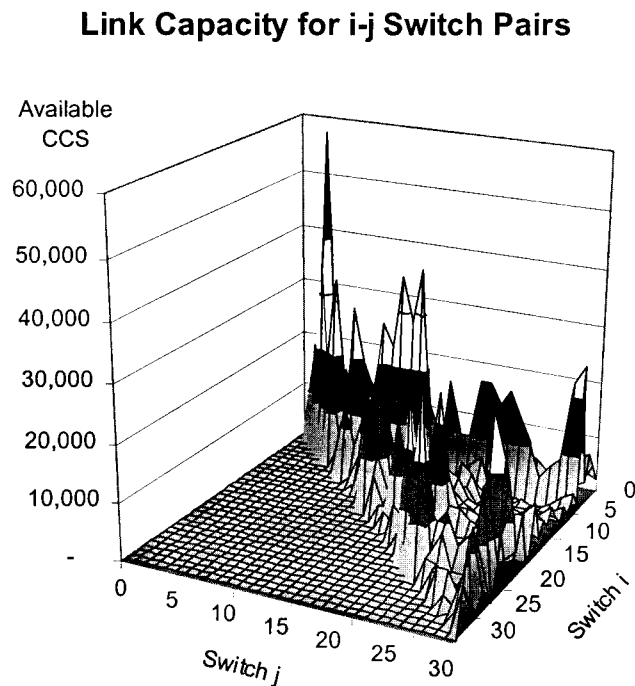
Figure 3 – Network capacity for July 31, 2000

Network Activity

In order to analyze the network, some sense of traffic behaviour was required. Call activity exhibits monthly, weekly and daily seasonality. The data storage requirements necessary to track monthly seasonality exceed available resources; therefore, weekly and daily seasonality will be discussed. Figure 4 shows call activity over a two-week period in November of 1999 in the network. Other than the drop in traffic over Remembrance Day (November 11th),[10] the four busiest days are Monday through Thursday (the *busy days*) with each day having three peaks. Approximately 21 million calls are processed on an

---

[10] Some holidays have the opposite effect, such as Mother's Day where call activity surges.

11

average busy day in the network. The weekends exhibit different traffic patterns: Saturday afternoons and Sunday evenings appear to be the busiest times.
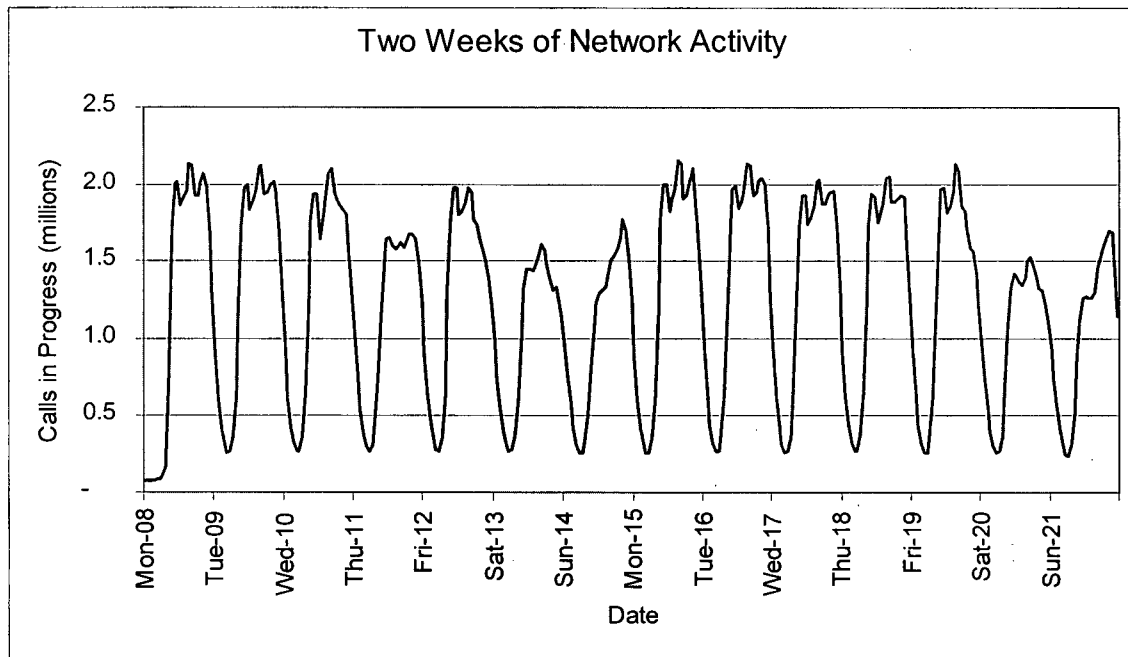
Figure 4 – Two weeks of traffic seasonality

Figure 5 shows the total CSS consumed over Monday, July 31, 2000. Figure 6 shows the total number of calls in progress over the same day. Notice that the number of calls in progress declines monotonically from 16:00 onwards, whereas the call volume actually rises in the late evening. This indicates an increase in average call duration, which was most likely due to an upsurge in Internet traffic and less expensive long distance calls. The above stated increase in average call duration was the impetus behind focusing on two data samples. The first being taken from the 11:00-12:00 time interval and the second from the 21:00-22:00 time interval.
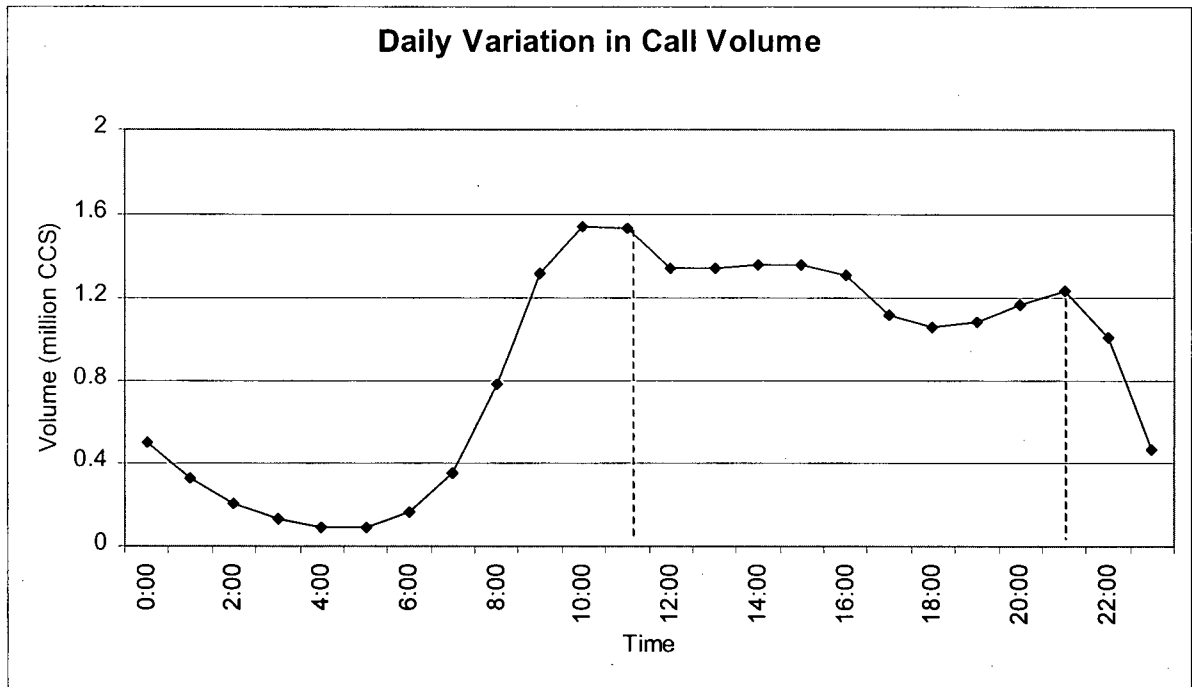
Figure 5 – An example of daily CCS seasonality
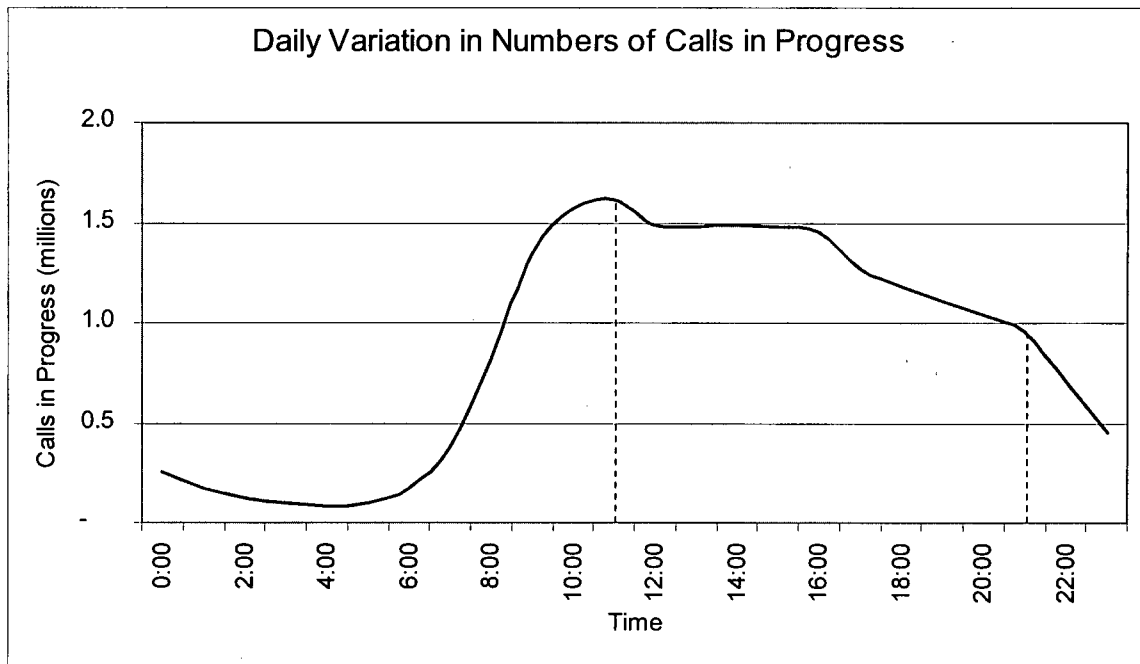
Figure 6 – An example of daily call seasonality

13

Modeling Considerations

Spatial data[11] was required for the network simulator (discussed chapter IV) and the optimization routines (discussed in chapter V). Based upon analysis of data samples from different times of the day, it appears that CCS carried on the network during the day was quite different from the night. The longer holding times present at 21:00 are supporting evidence. Consequently, a spatial data sample was collected over an hour during the day (11:00-12:00) and a second sample was collected over an hour at night (21:00-22:00).

Figure 7 and Figure 8 show the spatial data samples for the 11:00 and 21:00 time slots respectively. One of the more prominent changes was the rise in CCS between OD-pair 15-4 in the 21:00-22:00 time interval. Figure 9 illustrates the difference between the two samples, which reveals some similarities in traffic patterns between most OD-pairs. If the network has been dimensioned to handle traffic on the direct links for both scenarios, different pockets of capacity would be underutilized at the two different times. Consequently, time of day routing could diminish the need for network augmentation or curb it all together in the short term.

---

[11] The data was laid out spatially with network switches and trunks as the variables around which the information was gathered.

Demand Between OD-Pairs
11:00 - 12:00
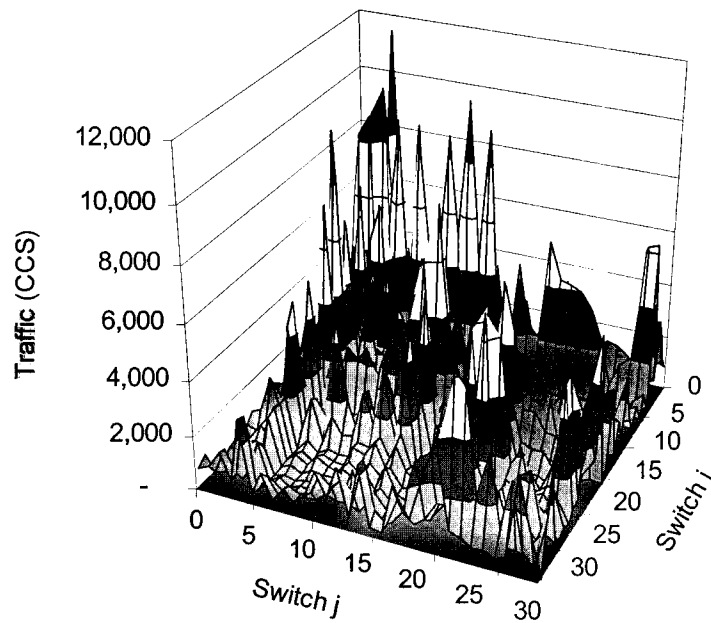
Figure 7 – A cross-sectional demand data sample at 11:00.

Demand Between OD-Pairs
21:00 - 22:00
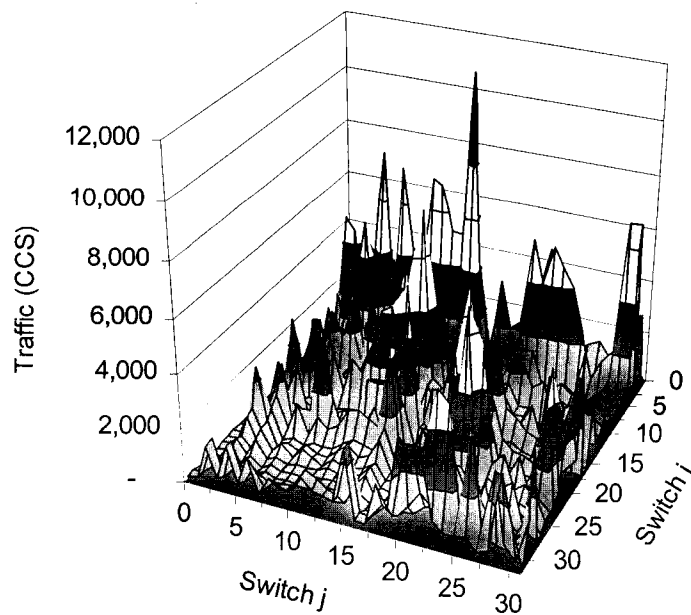
Figure 8 – A cross-sectional demand data sample at 21:00.

Difference Between OD-Pair Demand
(11:00-12:00 minus 21:00-22:00)
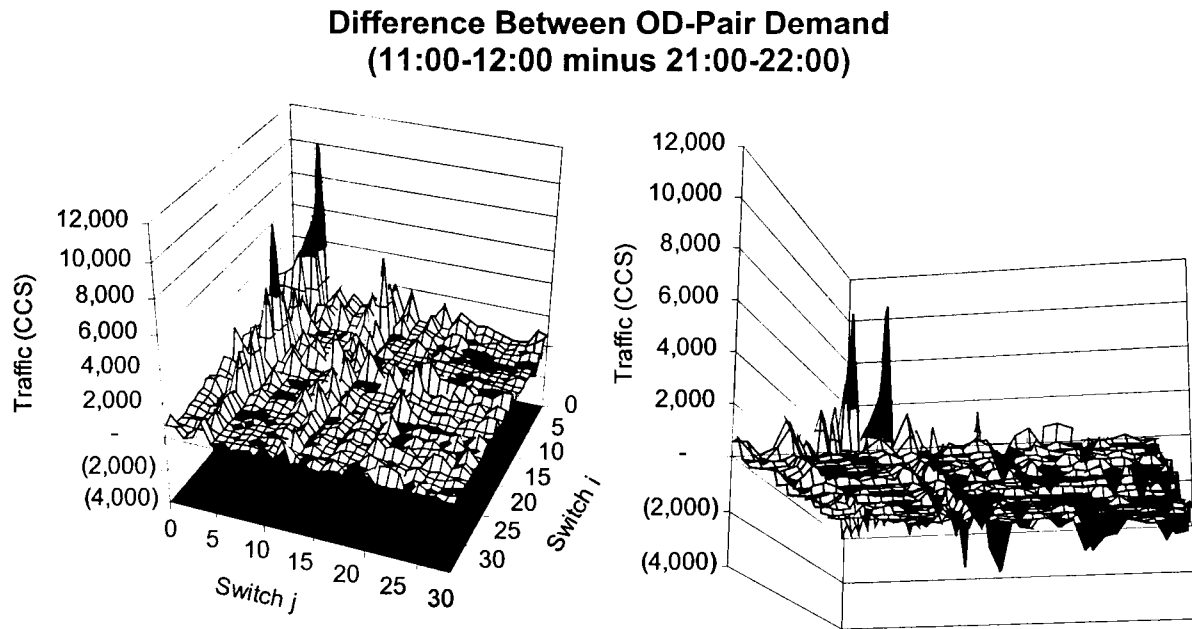
Figure 9 – Difference between 11:00-12:00 and 21:00-22:00 demand

Assumptions, Errors, and Limitations

Although it was a rich source of information, the required spatial demand data were not available in an immediately usable form. Consequently, data assembly and manipulation was performed. This required assumptions (specified below) and the assembly process was subject to computational constraints, which cumulatively affected data integrity.

Source data (records) were collected for each link in the network and contain information about the call origin and destination. Given that some routes traverse multiple disparate links, it was necessary to assemble records to estimate OD-pair demand. As a result, records that represent a segment of the same call were possibly left unmatched. For example, call records that should be assembled into a single record were collected from devices that monitor activity at mutually exclusive sets of switches. Furthermore, call records contain a call start time, which was one primary criterion used to match records,

16

based upon a clock local to each switch. The clock at a switch can drift and subsequently cause records that should be assembled to be modeled as separate calls. Moreover, OD-pair demand was historically unavailable; therefore, we have no basis for comparison of error. Thus, the order of magnitude of the overall error due to the data assembly process was difficult to measure.

The current AHR system uses routes with a maximum length of three links, but longer routes are possible. For example, once a call reaches its destination, say switch X, it can be redirected due to call forwarding to a cellular phone or messaging service. Once redirected, the call is effectively re-routed from X as if it were a new call. It will continue to occupy a circuit from the call origin to X, as well as from X to the new destination. The two portions of the call will be modeled as if they are independent events, when in fact they are not. On an average day, 25 million records will be generated from the network. The time required to process this large amount of data dictated that calls utilizing more than three links could not be modeled as one call. Such calls are modeled as two or more independent calls. For example, a $\rightarrow$ b $\rightarrow$ c $\rightarrow$ d $\rightarrow$ e could be the route for one call, but it could be modeled as one call from a $\rightarrow$ b $\rightarrow$ c and another call from c $\rightarrow$ d $\rightarrow$ e.

To recreate network activity, empirical distributions for call frequency (number of calls per second) and call duration (call holding time) were developed for each OD-pair. Calls originating outside the study network were assumed to start at the first switch they contacted. Similarly, calls terminating outside the study network were assumed to finish at the last switch they contacted in the study network. As long as entry points for outside calls remain the same, any results or recommendations will remain valid. In the event of change, one simply has to recalibrate the call frequency and call duration distributions.

Collecting data samples of network activity for this study was a resource intensive endeavor for both TELUS and COE personnel. Ideally, multi-day samples should have been used. However, the volume of data alone stipulated that a minimum of 24 working hours up to as many as 50 hours was required to

prepare one day's worth of information once it was received from TELUS. Consequently, single day samples of network activity were used to develop the call frequency and call duration distributions.

The final allowance for computational constraints pertains to the call duration distributions. The empirical probability distributions actually have heavy tails – a result of internet traffic being carried on the network. For memory management purposes, the call duration distribution was separated into two pieces. The bulk of the call duration distribution was modeled empirically and the tail was modeled theoretically.[12] The simulation draws from an empirical probability mass function for calls lasting up to 4000 seconds and from a theoretical shifted exponential distribution for longer calls.

Routing of Calls

Understanding how the network actually routes calls was important. At the heart of routing lies a routing table. To explain how calls are actually routed in the network, we will work through two examples. Table 1 shows the conceptual design of a routing table, which stores a sequence of routes associated with each OD-pair. Given that the network contains 32 switches, each one was assigned a number between 1 and 32 for use in the optimization routines.

OD-Pair		Routes		
Origin Switch	Destination Switch	Route 1	Route 2	Route 3
1	32	1, 32	1, 22, 32	1, 2, 32
24	15	24, 2, 15	24, 22, 15	
32	1	32, 1	32, 22, 1	

Table 1 – Part of a fictitious routing table[13]

For each OD-pair, the network will initially attempt to connect the call by route 1. If all circuits on route 1 are used, the network will try route 2. If all circuits on route 2 are occupied then it will try route 3. If

---

[12] Of the standard theoretical distributions tested, the research team found that a shifted exponential distribution fit the best. Work is currently under way to fit a heavy tailed theoretical distribution to the entire call duration distribution, which should further increase the accuracy of the simulation.

[13] The number of routes does not have to be restricted to 3.

all circuits on route 3 are used, then the caller receives a network busy signal. Table 1 is accurate in the sense that the number of routes per OD-pair does not have to be the same, that route 1 does not have to be direct, and that the routes for an OD-pair can be directionally different (e.g., for OD-pair 1-32 and OD-pair 32-1 can be different).

Example 1 – a blocked call due to network congestion

Let the routes for OD-pair 1-32 be those given in table 1 and let the number of trunks in link $\{1,32\}$ be 10. If all 10 circuits are occupied, then the system will attempt to complete the call via the second route $(1\rightarrow22\rightarrow32)$ in the routing table. If neither link $\{1,22\}$ nor link $\{22,32\}$ has trunks available, the system will attempt to connect via the third route $(1\rightarrow2\rightarrow32)$ in the routing table for OD-pair 1–32. If link $\{1,2\}$ or link $\{2,32\}$ is also full, then the caller will receive a fast busy signal indicating that the call was blocked due to network congestion.

Example 2 – a call rerouted via route 2

Let call X be initiated at a time when all trunks on the link $\{1,32\}$ are being used, but there are circuits available on the second route $(1\rightarrow22\rightarrow32)$. Again, the system will check if the call can be routed from 1 to 32 directly. In this scenario, there are no available circuits on route 1, so the call will overflow to the second alternate route. The system will attempt the second route $(1\rightarrow22\rightarrow32)$ and find at least one trunk free on link $\{1,22\}$ and one free on link $\{22,32\}$ (i.e., an available circuit on route 2). The system will then proceed to reserve a line on each link, thus creating a circuit for call X.

IV. THE NETWORK SIMULATOR

The simulation was based on work started last summer and other COE analysts have made continual improvements to it. The development of the network simulator is outside the scope of this thesis; however, we incorporated a widely used method to circumvent the unacceptably long warm-up period.

Thus, it warrants discussion. Furthermore, the simulator was the metric for determining the quality of results from the optimization routines.

A graphical depiction of the simulation is presented in Figure 10. The simulator requires the following inputs for each OD-pair: routing rules, a stationary call frequency distribution, and a stationary call duration distribution. The final input, network topology, indicates which switches are directly connected to each other, the corresponding link capacities, and trunk reservation parameters. Once all input files have been processed, the clock is started and the simulation enters the warm-up period. The purpose of the warm-up period is to prime the network until it reaches a steady state. Once the simulation has stabilized, performance is measured in order to assess the quality of the simulated network. Statistics are calculated both on a link and an OD-pair basis. Summary network statistics are then calculated from these detailed statistics.
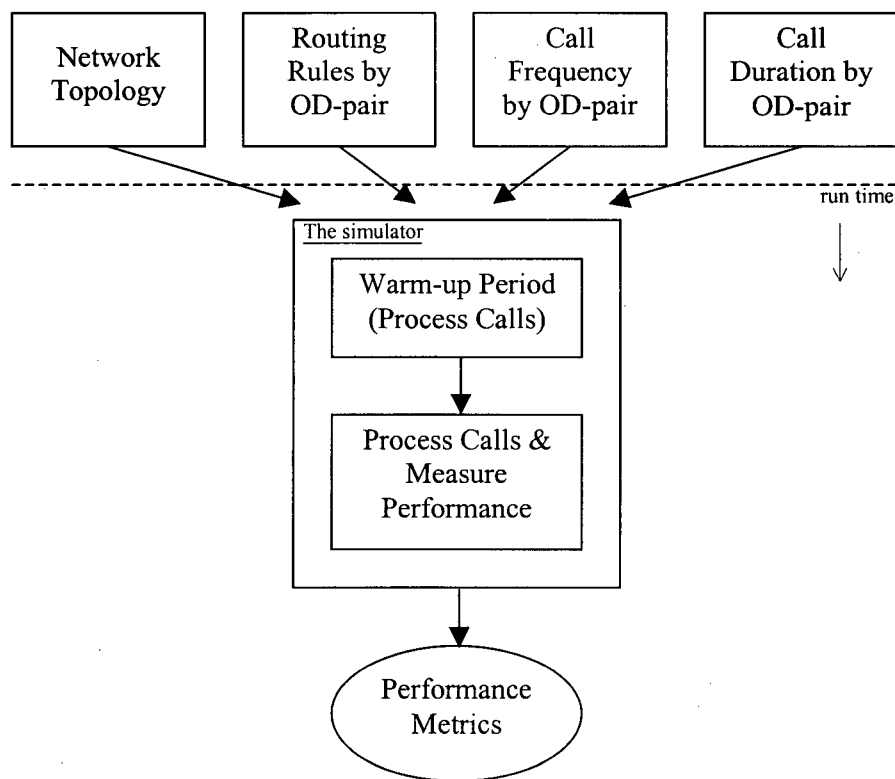
Figure 10 – A high level overview of the simulation

20

The Effects of Run Time upon Simulation Design

The duration of one replication used to take 21 simulated hours. The ratio of simulated hours to computer processing hours is approximately 5 to 3,[14] so 5 replications would require 63 hours of computer processing time in order to test a single scenario. This was unacceptably long; therefore, two avenues were explored to attempt to decrease runtime of the simulation: optimizing the warm-up period, and obtaining multiple output data samples within a single simulation run.

The initial analysis of the simulation was performed to ensure it matched reality. Therefore, effort was focused on the steady-state portion of the run time, and to be conservative, 20 simulated hours were used to prime the network. We implemented a widely used technique to determine the required warm-up period. From the simulation results, we concluded that *percent calls completed per minute* was the most variable statistic by comparing normalized variances of all outputs. Thus the time required for *percent calls completed per minute* to stabilize would be at least as long as all other output statistics. Consequently, it was used to calculate the minimum time required for the simulation to warm-up. From 6 replications, the time series of *percent calls completed per minute* were averaged together to generate a series with $1/6^{th}$ the variance of each individual series. A moving average was applied to smooth the aggregate series, which allowed for visual analysis of long term trends. Figure 11 shows the smooth time series that was used to determine that a warm-up period of 600 minutes (10 hours) was sufficient.

---

[14] On a Pentium III 800 with at least 128M of RAM
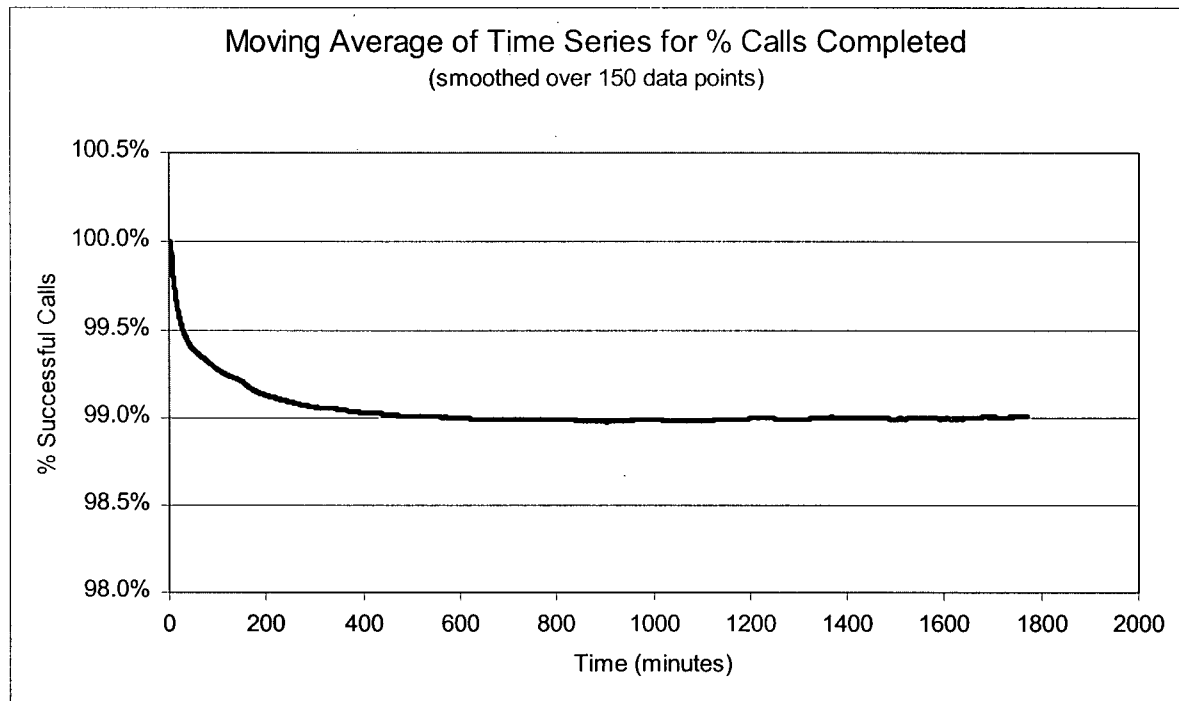
Figure 11 – The smoothed time series of % calls completed

After decreasing the warm-up period by a factor of two, the total simulation time of 11 hours (i.e., 6.5 computer processing hours) was still undesirable. If 5 replications were performed per scenario, approximately 32.5 hours of computational time would have been required. Multiple independent replications are an integral part of proper simulation design and therefore are a necessity. However, given the unacceptably long warm-up period, we wanted to incur this cost only once. The solution was to obtain multiple output samples from the same run of the simulation. The autocorrelation of call blockage per minute[15] was calculated to determine an acceptable length of time between sampling. The autocorrelation was within the 95% significance bounds after lag 100 (see Figure 12); therefore, the inter-sampling time could be 100 minutes.[16] Hence, the total computational effort has been reduced from 63 hours to 13 hours to collect 5 data samples.

---

[15] A statistic similar in behaviour to percent calls completed per minute
[16] The current inter-sampling time is 300 minutes.
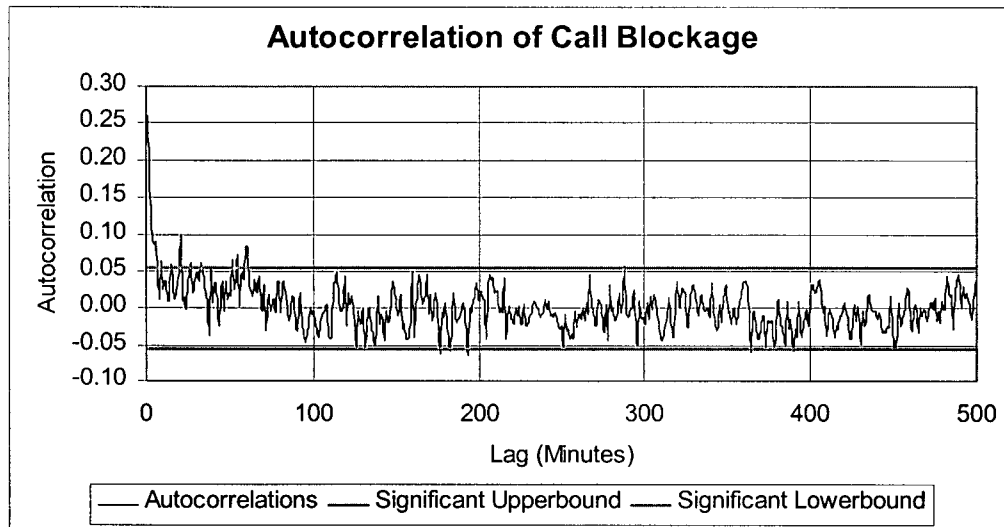
Figure 12 – Autocorrelation of percent call blockage in the network

Model Validation

Several metrics were used to determine whether the simulation of the network closely matches performance of the actual network: *trunk group utilization, overflow CCS, overflow calls, call attempts,* and *percentage of blocked calls.* The validation effort was undertaken by other members of the research team [4]; therefore it is outside the scope of this thesis. However, we will briefly discuss the two most important metrics, *percentage of blocked calls,* and *trunk group utilization.*

Comparing the percentage of blocked calls in the real network and the simulated network was rather trivial. Approximately 1 in every 6,400 calls and 1 in every 5,200 calls were blocked in the real network during the 11:00-12:00 and 21:00-22:00 time interval, respectively.[17] In other words, less than two one-thousandths of a percent of calls attempted were blocked. Approximately 1 in every 9,500 calls and 1 in every 8,900 calls were blocked in the simulated network. Given small number of blocked calls observed in all four cases, we were satisfied that the simulation was generating blockage similar to that observed in the network.

Overall, the simulated trunk group utilization was within 3% of the actual. Comparing on a trunk group basis, almost 50% of the groups were within 5%, and 92% of the groups were within 20%. It is important to note that the percent error measurement (|actual − simulated| / actual) is extremely sensitive to the size of the denominator; most of the large percent errors were associated with the smaller trunk groups. A more robust measurement is the difference in carried CCS. On average, the absolute difference was 200 CCS and the maximum was only 2900 CCS for the 11:00-12:00 time interval (less than two one thousandths of a percent of the total carried CCS). Another method used for validation was to visually compare the actual carried CCS to the simulated carried CCS on every trunk group (Figure 13). Each point in the graph compares the simulated to the actual carried CCS. A 45-degree line was added to the graph for a relative basis of comparison. As we can see, simulated trunk group activity was fairly representative.
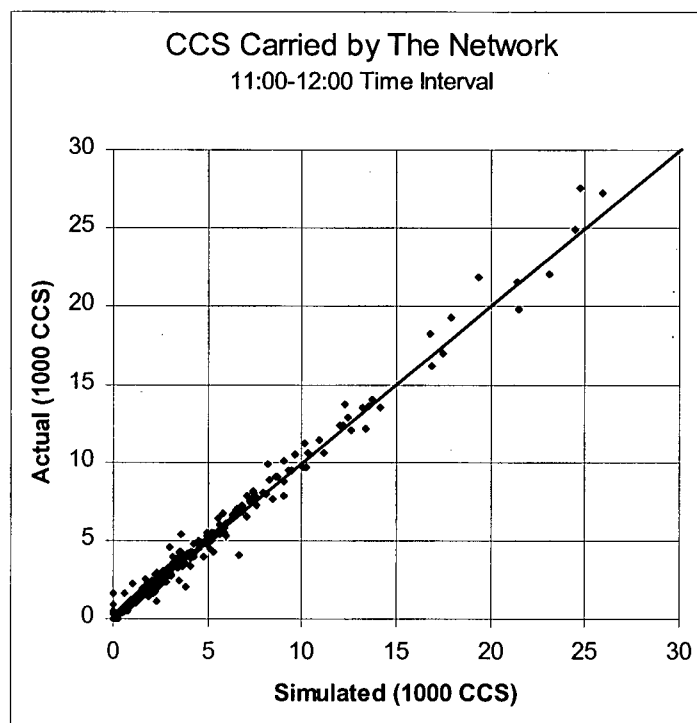
Figure 13 – Actual vs. Simulated CCS on Each Trunk Group

---

[17] Having such a low occurrence of call blockage was easily explained. The network was engineered to block 1 in every 1000 calls in the busiest hour of the busiest day of the busiest season. Since July 31, 2000 was not a day in the busy season for this network, few calls were blocked.

V.   ROUTING MODELS

Overview

The objective of developing routing models was to find new routing tables that improve network performance. If the routing table in each switch was changed, alternate hierarchical routing would most likely be replaced by alternate nonhierarchical routing (ANHR). Migrating to ANHR may increase the number of calls carried by taking advantage of non-coincident busy hours in different geographic areas of the network. To test this hypothesis, four mathematical programming models have been developed: a multi-commodity flow model, a mixed-integer program, a linear program, and a nonlinear program. Each model is introduced, formulated and evaluated in this chapter. Out of the four models, the most appealing was the 2-link maximum linear program and was subsequently used to generate scenarios discussed in chapter VI.

Math programming models were of primary interest for two reasons. Firstly, the size of the problem suggests that other alternatives, such as implementing a Markov Decision Process model or a stochastic linear program, would not be tractable. Secondly, it was of significant importance to TELUS to implement a solution as quickly as possible. Consequently, the simplicity offered by the developed math programming models allow for a quick response to TELUS' needs.

Model Parameters

All four math-programming models take the same two inputs: *trunks working* and the *average number of circuits required* for each OD-pair. To elaborate on the average number of circuits required, the models developed require non-probabilistic input; however, telephone network usage is inherently stochastic. Hence we simplified by engineering for the average case and assume that the average was representative of the underlying distribution. The *average number of circuits required* was calculated by multiplying the mean arrival rate by the mean call holding time for each OD-pair. The two matrices of OD-demand,

which were used as inputs to the models, are the source data to Figures 7 and 8 (see page 15). Generally, a third required parameter is *network connectivity* – information detailing which of the switches are directly connected to each other. However, since *trunks working* was stored for every pair of switches, the models knew which switches were connected to each other. Thus, network connectivity was inferred directly from the trunks working information. Consequently, it was not required as a separate input.

A) A Multi-commodity Flow Model (MCFM)

Overview

A model that distinguishes calls by origin and destination in order to trace call route(s) through the network was desired. This was achieved by treating each call origin as a separate commodity. Consequently, a multi-commodity network flow model (MCFM) was formulated by Queyranne [5] and programmed using XPRESS-MP.[18] The explicit objective was to minimize the maximum average link utilization while routing all demand between OD-pairs.[19] The implicit objective was to use the resulting flow variables to infer routes for each OD-pair. In the formulation, the traffic sent on each link has a set of flow variables; one for each type of commodity. After a run of the model, routes were found for each OD-pair by recursively tracing through the flow variables.

Variables

N	Set of nodes (network switches)
C_{jk}	The number of trunks in link $\{j,k\}$
D_{ik}	The average number of circuits required for each OD-pair i-k | i, $k \in N$ where $k \neq i$
x_{ijk}	= 0 if $C_{jk} = 0$ = the flow from i routed on link $\{j,k\}$[20] otherwise
λ	The maximum average link utilization

Table 2 – List of MCFM variables

---

[18] See Bertsimas, Tsitsiklis [1] for an introduction to multi-commodity flow models.
[19] It is "average" link utilization because demand between each OD-pair is the average number of lines required.
[20] This is not flow from switch i to j to k, rather flow on link $\{j,k\}$ that originated from i and has reached switch j.

Formulation

objective: $\quad \min \lambda$ $\hfill$ (1)

subject to: $\quad \displaystyle\sum_{j \in N} x_{ijk} = \sum_{l \in N} x_{ikl} + D_{ik} \quad \forall i,k \in N \; where \; j \neq k, \, i \neq k$ $\hfill$ (2)

$\displaystyle\sum_{i \in N} (x_{ijk} + x_{ikj}) \leq \lambda \, C_{jk} \quad \forall \, j,k \in N \; where \, k > j, \, i \neq j, i \neq k$ $\hfill$ (3)

A MCFM requires a flow balance equation (2), which can be conceptualized as "flow in" = "flow out". The flow variable X_{ijk} represents the flow originating from switch i that has entered switch k from switch j. Next, the flow variable X_{ikl} represents the flow originating from switch i that has left switch k and is entering switch l. Finally, the demand D_{ik} draws flow from the origin switch i to destination switch k in order to satisfy the equality in the constraint (see Figure 14).
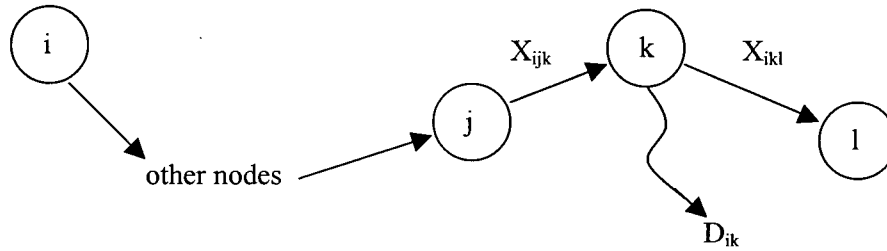
Figure 14 – Illustration of the flow variable in the multi-commodity flow model

Objective functions can vary widely and give the developer some degree of autonomy in MCFM's. A common technique in optimization is to add a multiplier to a right hand side (RHS) of some constraints and then minimize the multiplier. Given no other suitable objective (e.g., minimizing cost), we minimized a multiplier (1) and it was included in the RHS of the capacity constraints (3). Thus, this model minimizes the maximum average link utilization.

Inferring Routes from Flow Variables

After the MCFM has been solved, the flow variables (x_{ijk}'s) contain link-specific information. The flow variables were analyzed to determine which route(s) each commodity used by tracing traffic on adjacent

links from each destination back to each origin. Once all routes were traced, the final step was to determine the order in which the routes should be attempted in the simulator for each OD-pair. Routes were ranked for each OD-pair based upon the average number of circuits used on each route and the ordered sets were subsequently stored in a routing table.

Discussion

The model was run on several data sets (see Appendix 1 for a sample data set) and the optimal solutions generated unappealing sets of routes for some OD-pairs (see Appendix 2 for a part of a sample solution). Several pairs had no alternate routes and others had up to as many as 47. Finally, the post processing effort required a considerable amount of human intervention. Consequently, an attractive alternative was to develop a model that restricted or minimized route length across all OD-pairs. Alterations to balance and minimize route length were undertaken and were the impetus for the next model developed.

B) A Mixed-integer Program

Overview

The MCFM did not suggest a good set of routes for all OD-pairs. It favoured shorter sets of routes for certain pairs at the expense of longer routes for other pairs. Consequently, the model was extended in an attempt to address this undesired feature of the MCFM solutions. We decided to minimize the maximum number of links used by each commodity. This would in essence balance the route lengths across all commodities at the computational expense of introducing binary variables. Minimizing the maximum average link utilization was still important so combining both this new objective with the original MCFM objective into a single formulation seemed appropriate.

Variables

N	The set of nodes (network switches)
C_{jk}	The number of trunks in link {j,k}
D_{ik}	The average number of circuits required for each OD-pair i-k | i, k∈ N where k ≠ i
x_{ijk}	= 0 if C_{jk} = 0 = the flow from i routed on link {j,k} otherwise
y_{ijk}	= 1 if link {j,k} is used by commodity i = 0 otherwise
λ	The maximum average link utilization

Table 3 – List of integer program variables

Formulation

conceptual objective: $\min \max_{i \in N} \left(\sum_{j \in N} \sum_{k \in N} y_{ijk} \right)$ (4)

objective: $\min z$ (5)

subject to:

$$\sum_{j \in N} \sum_{k \in N} y_{ijk} < z \quad \forall\, i \in N \; where\, j \neq k, i \neq k \tag{6}$$

$$(x_{ijk} + x_{ikj}) \leq y_{ijk} C_{jk} \quad \forall i,j,k \in N \; where\, k > j, i \neq j, i \neq k \tag{7}$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{ikj} + D_{ik} \quad \forall i,k \in N \; where\, j \neq k, i \neq k \tag{8}$$

$$\sum_{i \in N} (x_{ijk} + x_{ikj}) \leq \lambda\, C_{jk} \quad \forall j,k \in N \; where\, k > j, i \neq j, i \neq k \tag{9}$$

Minimizing the maximum number of links used by each commodity was a piecewise linear and convex objective (4). However, it was modeled linearly by changing the objective function to a constraint and minimizing the right hand side (5) and (6).[21] One additional constraint was necessary to ensure the y_{ijk} variables were activated when a link {j,k} was used (7).

Equations (8) and (9) are identical to the flow balance constraints (2) and the capacity constraints (3), respectively, form the MCFM.

---

[21] An introduction to this technique can be found in a text by Bertsimas and Tsitsiklis [1].

To incorporate the original MCFM objective, a weighted objective function could be used (10). In the event that one objective appears to dominate the results, a multiplicative weight could be added to the other term.

alternative objective: $\min \lambda + z$ (10)

Discussion

When initially developing the model, we scaled down the problem from the 32-node network to a 5-node fully connected sample network. The purpose was to ensure the model was programmed correctly before consuming large amounts of run time. The 5-node mixed integer problem solved in approximately 6.5 hours compared to the three minutes required for the 32-node MCFM to solve on a Pentium III 600 with 256MB of RAM. Given this result, further development of mixed integer models was not pursued. The introduction of the binary variables produced unacceptable runtimes even in small instances. This result was the primary impetus for development of the next linear program.

C) A 2-link Maximum Linear Model (2MLM)

Overview

The MCFM suggested long routes and the mixed integer model required too much time to solve. The development of this linear program addressed the undesirable traits of both models by limiting the alternate routes to a maximum of two links in length. Like the MCFM, the 2-link maximum linear model (2MLM) was formulated by Queyranne [5], and it was developed in AMPL to generate inputs for CPLEX.

Variables

N	The set of nodes (network switches)
C_{jk}	The number of trunks in link {j,k}
D_{ik}	The average number of circuits required for each OD-pair i-k | i, k∈N where k ≠ i
z_{ik}	= 0 if C_{ik} = 0 = the average number of circuits used on route (i→k) otherwise
z_{ijk}	= 0 if C_{ij} or C_{jk} = 0 = the average number of circuits used on route (i→j→k) otherwise
λ	The maximum average link utilization

Table 4 – List of linear program variables

Formulation

objective: $\quad \min \lambda$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ (11)

subject to: $\quad z_{ik} + \sum_{j \in N} z_{ijk} = D_{ik} \quad \forall i,k \in N \ where \ i \neq j, i \neq k, j \neq k$ $\qquad$ (12)

$$z_{uv} + z_{vu} + \sum_{i \in N}(z_{iuv} + z_{ivu} + z_{uvi} + z_{vui}) \leq \lambda \, C_{uv} \quad \forall \, u,v \in N \ where \ v > u \qquad (13)$$

The 2MLM incorporates the same technique used in the MCFM of minimizing the right hand side multiplier of the capacity constraints (11). All demand has to be satisfied (12) and traffic sent on a link cannot exceed the number of trunks working (13).

Model Extension

Aggregating Routes

The above formulation permitted the solver to select different routes for flow from switch i to switch k and for flow from switch k to switch i. However, this formulation did not contain sufficient information to determine that traffic in each direction was of equal value. Consequently, all flow from one direction was routed directly while the flow in the opposite direction was forced onto 2-link routes. Table 5 shows the model's suggested routes for switches 0 and 1, which is a typical example of the routes for other OD-pairs.

Switches			Circuits
Origin	Tandem	Destination	Used
0	9	1	350
0	19	1	344
0	28	1	259
0	24	1	211
1		0	1,516

Table 5 – Routes for OD pairs 0-1 and 1-0

To obtain balanced sets of routes in both directions equation (12) was reformulated. The *average number of circuits required* between switch k and switch i was added to the *average number of circuits required* between switch i and switch k (14). Thus, half as many flow variables were created. The interpretation of z_{ik} changed to indicate the flow on route (i→k) plus the flow on route (k→i). Similarly, z_{ijk} indicated the flow on route (i→j→k) and route (k→j→i) combined. Consequently, this revised model suggests the same set of routes for both OD-pairs i-k and k-i.

$$z_{ik} + \sum_{j \in N} z_{ijk} = D_{ik} + D_{ki} \quad \forall i, k \in N \ where \ i \neq j, j \neq k, k > i \tag{14}$$

Inferring Routes from Flow Variables

Each OD-pair had a set of flow variables associated with it, and these variables directly corresponded to routes for each OD-pair. Hence, no effort was required to infer routes from the flow variables. Given that the sum of all flow variables must equal the demand expressed in average number of circuits required, it seemed logical to rank the routes according to the average number of circuits required. Consequently, the variable with the highest average number of circuits required would imply that the route associated with it would be attempted first. The variable with the second highest average number of circuits required would be attempted next. This process continued until all flow variables were ranked for each OD-pair. Table 6 details the ranking for OD-pairs 1-4 and 4-1 during the 11:00-12:00 time slot.

Variable	Flow	Route Ranking	Routes for 4-1	Routes for 1-4
X_{104}	41.4	2	4→0→1	1→0→4
X_{14}	343.5	1	4→1	1→4
X_{124}	28.9	3	4→2→1	1→2→4
Total flow:	413.8	D_{ik}:	413.8	

Table 6 – Routes for OD-pairs 4-1 and 1-4 from the 11:00-12:00 time interval.

Discussion

The 2MLM solves in less time than the MCFM developed in XPRESS-MP. Furthermore, migrating to the 2MLM to CPLEX decreased the total runtime to approximately 30 seconds. As the name implies, the 2MLM restricts route lengths to a maximum of two links. Thus, the two objectives set out for this model – restricting route length and solving quickly – were achieved. An additional benefit was that each route has a variable associated with it. Thus, the average number of circuits used on a specific route was directly inferred from the value of the corresponding flow variable. Consequently, a routing table was developed with little post processing. A final benefit was that the model could be extended to include routes of longer length. For example, if TELUS expressed interest in solutions with three link routes, the model could be reprogrammed in a matter of minutes.

D) A Nonlinear Model

Overview

The 2MLM assumes a linear relationship between demand offered to a link and the demand carried on a link. Work dating back to Erlang demonstrates that the relationship is in fact nonlinear. To incorporate this relationship, a model with a nonlinear, convex objective function and linear constraints was developed. The move to nonlinear models generally implies a significant increase in runtime. However, one set of constraints in the 2MLM is not present in the nonlinear model. As will be shown shortly, the capacity constraints are now incorporated in the objective function. Thus, the runtime only increased from 30 seconds in the 2MLM to approximately 8 minutes in the nonlinear model.

The objective function was convex and the constraints were linear. Consequently, convex programming methods could have been used; however, we decided to use a nonlinear solver because it was readily available for use. Open source code programmed in Fortran for MINOS – the available nonlinear solver – was used to develop a model suggested by Girard [2]. To invoke the model, we programmed the nonlinear convex objective function and recompiled the source code. Finally, a matrix generator was written in C to produce the text files required by the solver.

Variables

N	The set of nodes (network switches)
C_{jk}	The number of trunks in link {j,k}
D_{ik}	The average number of circuits required for each OD-pair i-k	i, k∈ N where k ≠ i
z_{ik}	The quantity of i-k demand routed on the direct link otherwise
z_{ijk}	The quantity of i-k demand routed through tandem j otherwise
x_{ij}	The total offered traffic to link {i,j}

Table 7 – List of nonlinear program variables

Formulation

objective: $\quad \min \sum_{i\in N}\sum_{j>i} x_{ij} * B(x_{ij}, C_{ij})$ $\hspace{3cm}$ (15)

subject to: $\quad z_{ik} + \sum_{j\in N} z_{ijk} = D_{ik} + D_{ki} \quad \forall\, i, k \in N \text{ where } i \neq j,\, j \neq k, k > i$ $\hspace{1cm}$ (16)

$$z_{ij} + z_{ij} + \sum_{k\in N}(z_{ijk} + z_{kij} + z_{jik} + z_{kji}) = x_{ij} \quad \forall\, i, j \in N \text{ where } i \neq k, j \neq k, j > i \quad (17)$$

$$with\ B(x_{ij}, C_{ij}) = \frac{x_{ij}^{C_{ij}} / C_{ij}!}{\sum_{i=0}^{C_{ij}} \frac{x_{ij}^{C_{ij}}}{i!}} \hspace{3cm} (18)$$

The Erlang-B loss function (18) is widely used to model the relationship between offered and blocked traffic on a trunk. Given the number of trunks on a link and an offered load (i.e., average number of circuits required), the Erlang-B function returns the proportion of blocked calls on the trunk. The objective of this model was to minimize the total expected number of blocked calls in the network (15). This was achieved by multiplying the traffic offered to link {i,j} by the proportion of calls blocked. The

resulting objective function was convex, which guarantees that the model converged to the optimal solution of the formulation. Equation (17) indicates how the offered traffic x_{ij} to link {i,j} was determined and equation (16) ensures that all demand was routed from its origin to its destination.

The Erlang-B Approximation

To invoke the model formulated above, some methods calculate link blockage by recursively looping over trunks working or offered traffic on each link. Instead of complete enumeration, other methods approximate the Erlang-B function because the number of trunks working and the amount of offered traffic is too large. Both methods assume an integer value for offered traffic and situations arise where the offered traffic to a link is fractional, which was true in this case. We decided to use an iterative method to calculate the carried link load and we used adjustments to incorporate the non-integer offered traffic. This entire procedure was taken directly from [4]. To start, the method uses a recursive equation that calculates the proportion of blocked calls given the offered traffic (19) and it was initialized by (20).

$$B(x_{ij}, C_{ij}) = \frac{x_{ij} B(x_{ij} - 1, C_{ij})}{x_{ij} B(x_{ij} - 1, C_{ij}) + x_{ij}} \tag{19}$$

$$where\ B(1, C_{ij}) = 1 \tag{20}$$

Equations (19) and (20) were programmed into the objective function and are only accurate for integer values of x_{ij}. However, D_{ij} was actually non-integer, and therefore, at least some x_{ij}'s were non-integer. Thus, equations (21) and (22) were programmed into the objective function to approximate blockage given the non-integrality of x_{ij}. As suggested in [4] the following approximation was used in the objective function:

$$B(x_{ij}, C_{ij}) \cong B^{1-h} B_1^h \left[\frac{B_1^2}{BB_2} \right]^{\frac{1}{2}h(1-h)} \tag{22}$$

$$where\ \ B = B(n, C_{ij}),\ \ B_1 = B_1(n, C_{ij}),\ \ B_2 = B_2(n, C_{ij})$$
$$and\ n = the\ closest\ integer\ to\ x_{ij}$$

MINOS is capable of estimating gradients used to determine the search direction for the next iteration. When MINOS estimated gradients, a 6-switch test problem was solved within a few minutes; however, the 32-node network model running on a Pentium III 800 with 256MB of RAM ran for an entire day and was still on the first iteration.[22] Consequently, (23) was programmed into the objective function to estimate the rate of change in blockage with respect to offered load to each link. The model subsequently solved in approximately 8 minutes after incorporating this gradient estimation.

$$\frac{\partial B(x_{ij}, C_{ij})}{\partial x_{ij}} = \left(\frac{C_{ij}}{x_{ij}} - 1 + B(x_{ij}, C_{ij}) \right) B(x_{ij}, C_{ij}) \tag{23}$$

Discussion

Programming of the matrix generator and the nonlinear objective function in Fortran for use in MINOS was a time consuming process. Consequently, the flexibility of the model was quite limited. For example, the replacement of (12) with (14) in the 2MLM required a short time to reprogram, whereas the same replacement in the nonlinear model required a relatively large amount of rework and debugging.

The nonlinear model overestimated blockage because the amount of offered traffic to the second link on a 2-link route was not thinned[23] by the blockage rate on the first link. Therefore, direct links were more preferred over 2-link routes relative to the 2MLM. Since the traffic offered to the second leg of a 2-link route is not thinned, the model overestimates the number of blocked calls. Without this assumption, nonlinear constraints would have been required to thin the traffic and it was unclear as to whether this new set of constraints was convex. This nonlinear model was developed as a stepping stone to a more complex nonlinear model that would have more accurately accounted for blockage. Pursuance of these nonlinear models completed when it was determined that the long-term objective of the research project with TELUS was to develop models that included trunk reservation. However, if the results from the

---

[22] The model was terminated before completion because any runtime over a few hours was unacceptable.

2MLM did not considerably decrease blockage in the simulated network, then this model or possibly a more complicated nonlinear model would have been revisited to unearth a set of routing rules for TELUS in the short-term.

VI.   SCENARIOS

Approximately 1040 hours of computing time were used to generate the 52 scenarios presented in this section. All scenarios were used to compare the current operation of the network with scenarios based upon routing rules suggested by the 2-link maximum linear model (2MLM). Of particular interest was the call blockage generated by the routing rules from the linear program in comparison to routing rules currently used by TELUS. To compare scenarios, we scaled up call attempts uniformly across the network, and tested the sensitivity of results to different reservation parameters. Finally, the benefit of time of day routing was assessed.

Routing Rules Generated by The Linear Program

Testing new sets of routing rules on a simulation with an extremely low percentage of blocked calls would not allow any performance improvements to be distinguished. Consequently, the number of calls attempts between each OD-pair was scaled up by 50% to compare the sets of rules. Figure 15 shows the mean (as well as its 99% confidence interval) number of calls blocked in the network when routing rules suggested by the 2MLM and TELUS' current hierarchical rules were used in the network simulator. Clearly, the hierarchical routes generated half the number of blocked calls as the routes suggested by the 2MLM. This was expected and the methodology for extracting value from the 2MLM results is discussed next.

---

[23] When blockage occurs on the first link, then less traffic would be offered to the second link in the real network.

	% Calls Blocked for The Status Quo and Linear Program Routes (50% Increase in Call Demand)	
	Status quo	Linear program
▬ 99% upper bound	0.0524	0.1082
⊓ blockage	0.0512	0.1075
▬ 99% lower bound	0.0501	0.1067
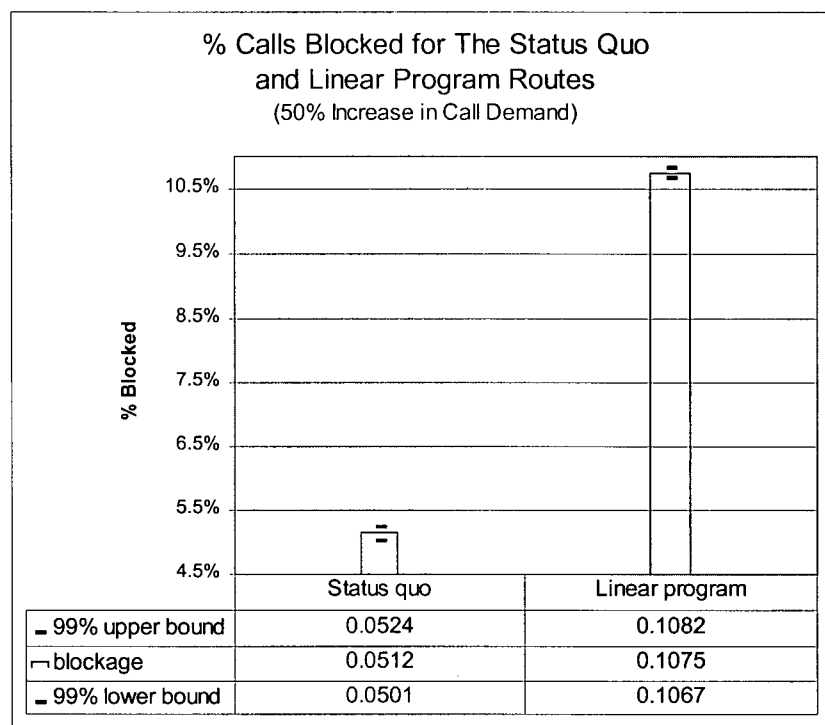
Figure 15 – The performance of 2MLM routes versus the currently used routes

Modifications to The 2MLM Routing Rules

The performance of the 2MLM's routing rules were as expected. Given its objective to minimize the average number of circuits used and that 2-link routes were twice as expensive as 1-link routes, the 2MLM intuitively uses 2-link routes sparingly. In essence, the model only rerouted traffic through under-utilized trunks. Consequently, the 2MLM did not suggest any alternate routes for most of the OD-pairs (see table 8). This was problematic because the network was engineered for alternate routing, and therefore, the direct routes are not dimensioned to carry all traffic between each OD-pair.

The value in the 2MLM's routes lie in the fact that the model seeks out underutilized links and uses them in 2-link alternate routes. With the intention of decreasing the number of blocked calls in the network, it was decided that the best alternative was to marry TELUS' current routing rules with the rules suggested by the 2MLM. This seemed appropriate because TELUS' rules incorporate hierarchical routing for which the network was designed, and the 2MLM routes seek out underutilized links.

In the new routing table, it was unclear as to whether the hierarchical routes or the 2MLM should be attempted first. For each OD-pair, the 2MLM routes could be attempted first followed by the current *alternate* routes. On the other hand, the current routes could be attempted first followed by the 2MLM routes. A third combination could have the direct route first, followed by the 2MLM routes with the current *alternate* routes attempted last. Thus, three sets of routing rules, one corresponding to each alternative, were generated. The first set had the 2MLM routes placed above the current alternate routes in the routing table and the second set had the order reversed. The third set was similar the first set except a direct route was attempted first, followed by the 2MLM routes, and then the current *alternate* routes.[24] Figure 16 shows the simulated total number of blocked calls of these routing rules compared with the status quo routes at a 50% increase in demand for the 11:00-12:00 data sample.
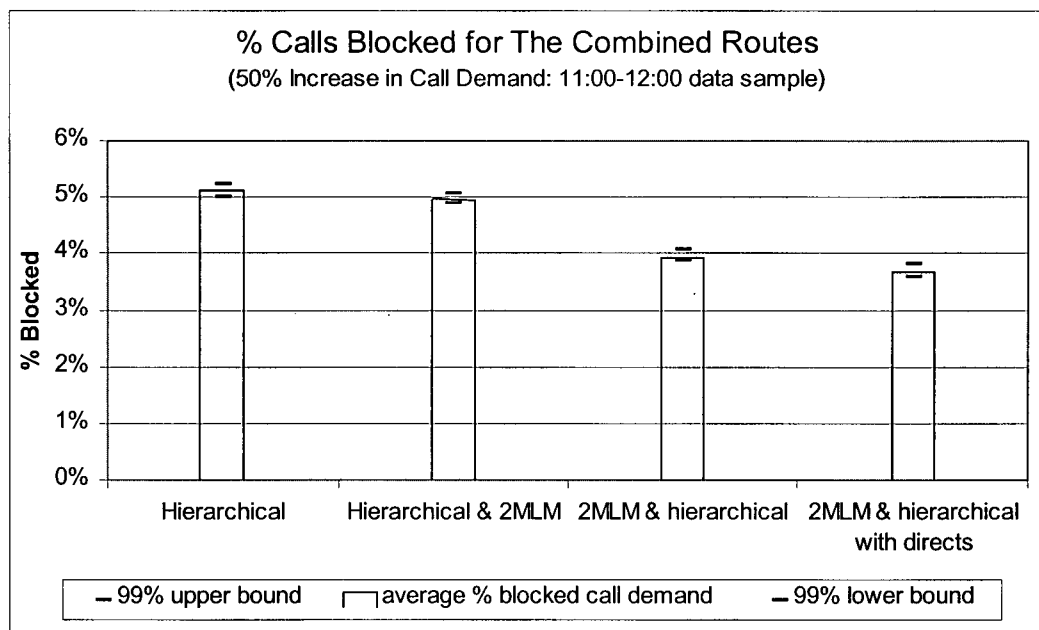
Figure 16 – Percentage of blocked calls for the combined route scenarios
for the 11:00-12:00 time interval

The rules where the 2MLM routes were attempted first performed better than the status quo; blockage was decreased by 1%. The performance of the reversed set of routes rules (i.e., current hierarchical routes

---

[24] Whenever the 2MLM suggested use of the same routes as the current routes, the second entry in the routing table was simply removed.

first) was relatively unimpressive. Consequently, further sensitivity analysis and scenarios with the reversed set was not pursued.

Similarly, Figure 17 shows the performance of 2MLM routes optimized for the 21:00-22:00 scenario. However, in this case the hierarchical routing rules performed twice as well as the combined hierarchical and 2MLM routing rules. A 3-fold performance improvement was achieved from a new routing table, which had the direct route first, followed by the 2MLM routes, and then the alternate hierarchical routes.
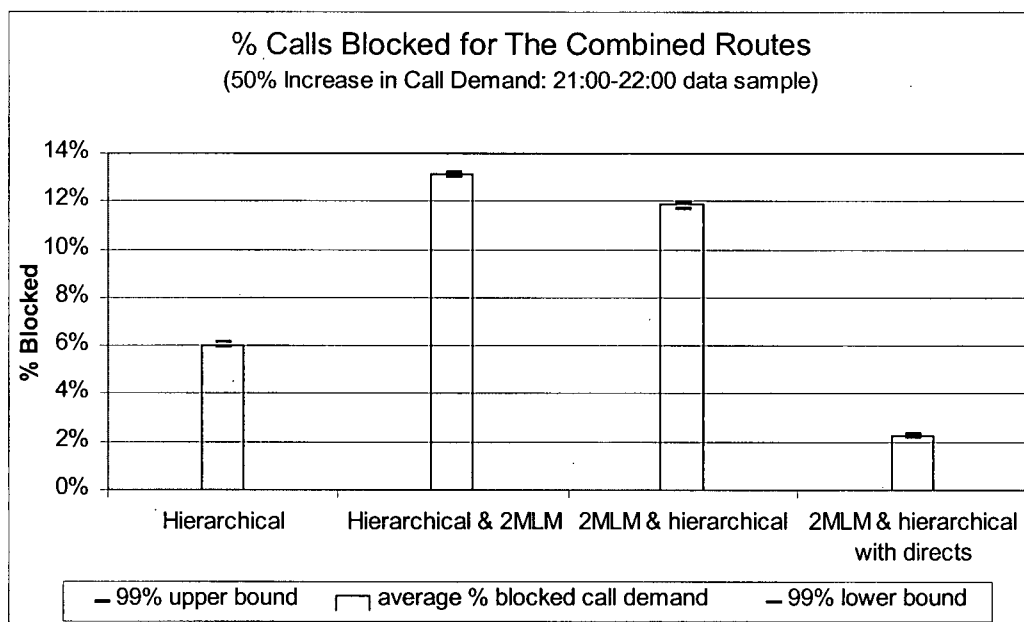
Figure 17 – Percentage of blocked calls for the combined route scenarios
for the 21:00-22:00 time interval

Table 8 summarizes the different routing rules that were tested. Given their performance, the routing rules with the most direct connections outperform the others. Furthermore, routing rules with 40% to 60% more alternate routes than direct routes outperform those with fewer alternate routes.

	Status Quo	11AM 2MLM Rules	Modfied 2MLM Rules		2MLM Rules with Direct Route	
			11 AM	9 PM	11 AM	9 PM
1-link Routes	918	830	830	745	918	918
2-link Routes	930	474	1,260	1,379	1,260	1,502
3-link Routes	442	-	424	390	424	424
Total	2,290	1,304	2,514	2,514	2,602	2,844
Avg. / OD-pair	2.34	1.36	2.63	2.63	2.72	2.97

Table 8 – Routing rule summary for three scenarios

The Effects of Scaling Call Attempts upon Network Performance

In order to assess the relative performance of the combined routing rules, demand placed upon the network had to be increased to create blockage in the network. The number of calls attempted between each OD-pair was scaled upon by 10%, 20%, 30%, 40% and 50%. The mean of the call arrival distribution for each OD-pair was scaled up and used in a theoretical Poisson distribution to generate new call duration distributions for each OD-pair. Figure 18 illustrates the performance of three sets of routing rules on the 11:00-12:00 time interval for varying levels of call attempts. Both sets of combined routing rules outperform the hierarchical rules. The combined routes with the direct link attempted first performs marginally better than the combined rules without the direct. Overall, these two sets of routing rules differed by 88 direct routes that were added.
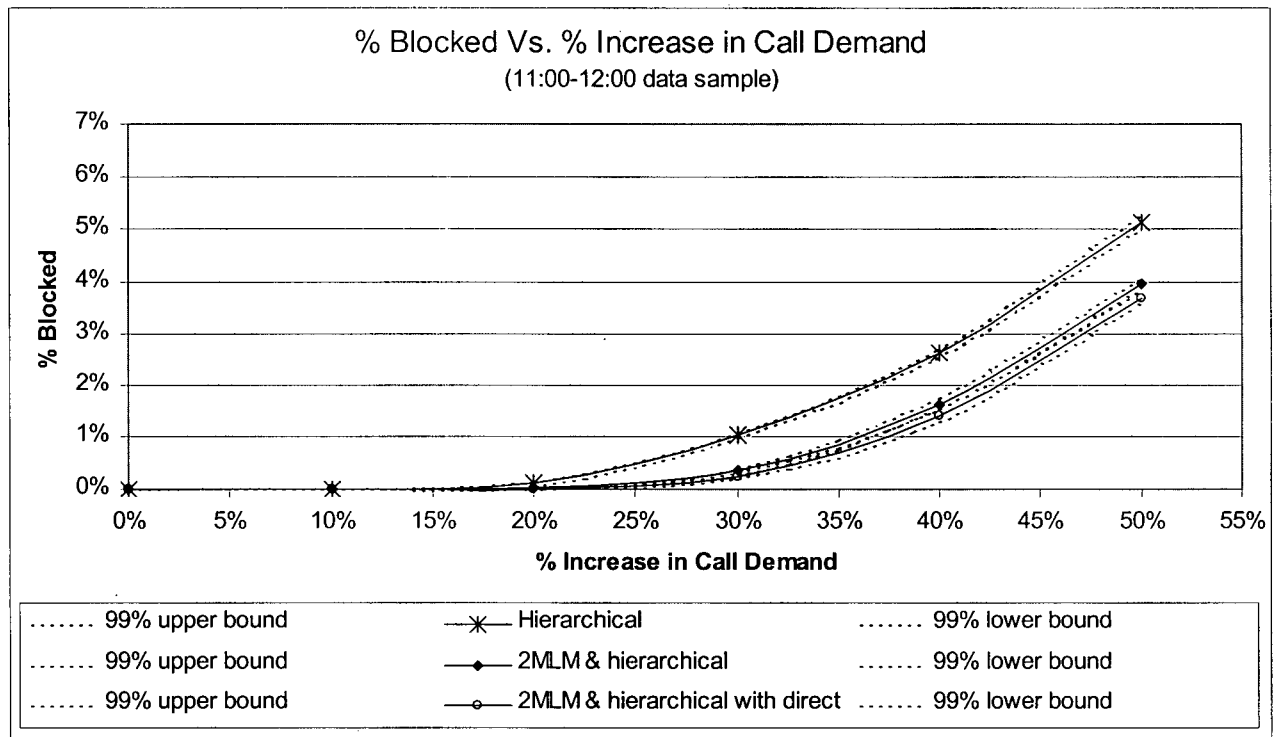
Figure 18 – The effect of increasing call attempts upon blockage for the 11:00-12:00 time interval

A validity check of the network simulator was to compare blockage to link utilization. Since the percentage of blocked calls decreased, more calls were carried by the network. Consequently, the utilization of the links in the network should increase. Figure 19 displays percent idle and percent utilized CCS summed over all links in the network. Note the increased utilization for the network for the combined scenario.
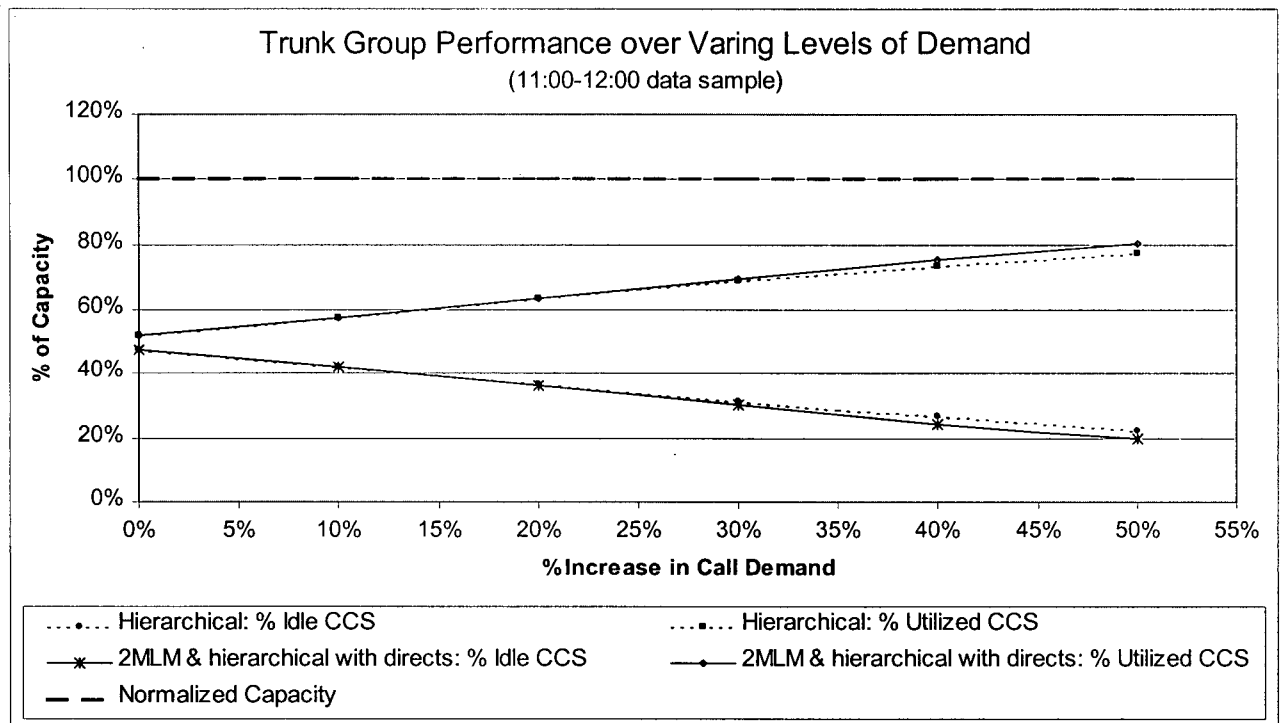
Figure 19 - The effect of increasing call attempts upon utilized and idle CCS for the 11:00-12:00 time interval

Figures 20 and 21 illustrate that similar behaviour occurred during the 21:00-22:00 time interval. The primary difference was that the relative improvement from use of the combined routing rules with direct routes added was more pronounced. At a 50% increase in call attempts, the percentage of blocked calls was approximately one third of the percentage that the hierarchical rules generates. The performance of the 2MLM married with the hierarchical routing rules over varying levels of call attempts was not generated because of their poor performance. 173 direct routes were added in total to the combined 2MLM and hierarchical routing rules (relative to 88 for the 11:00-12:00 time interval), which possibly explains the discrepancy in performance.
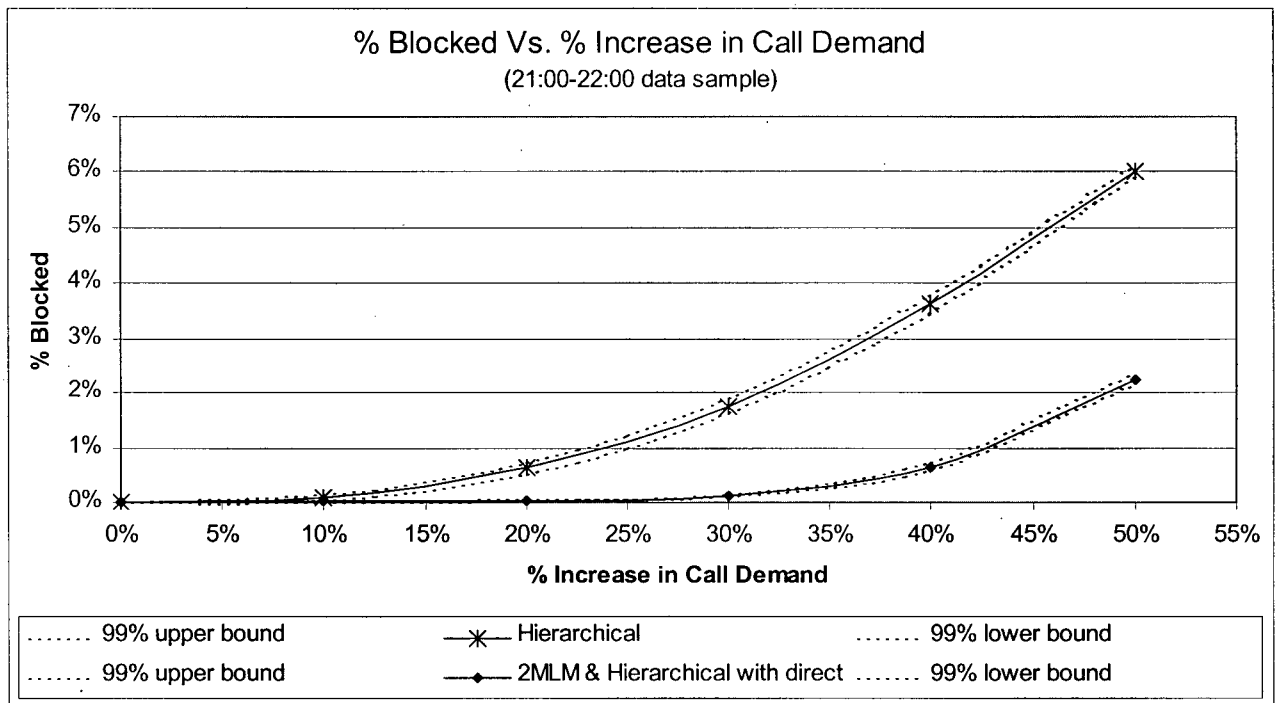
Figure 20 – The effect of increasing call attempts upon blockage for the 21:00-22:00 time interval
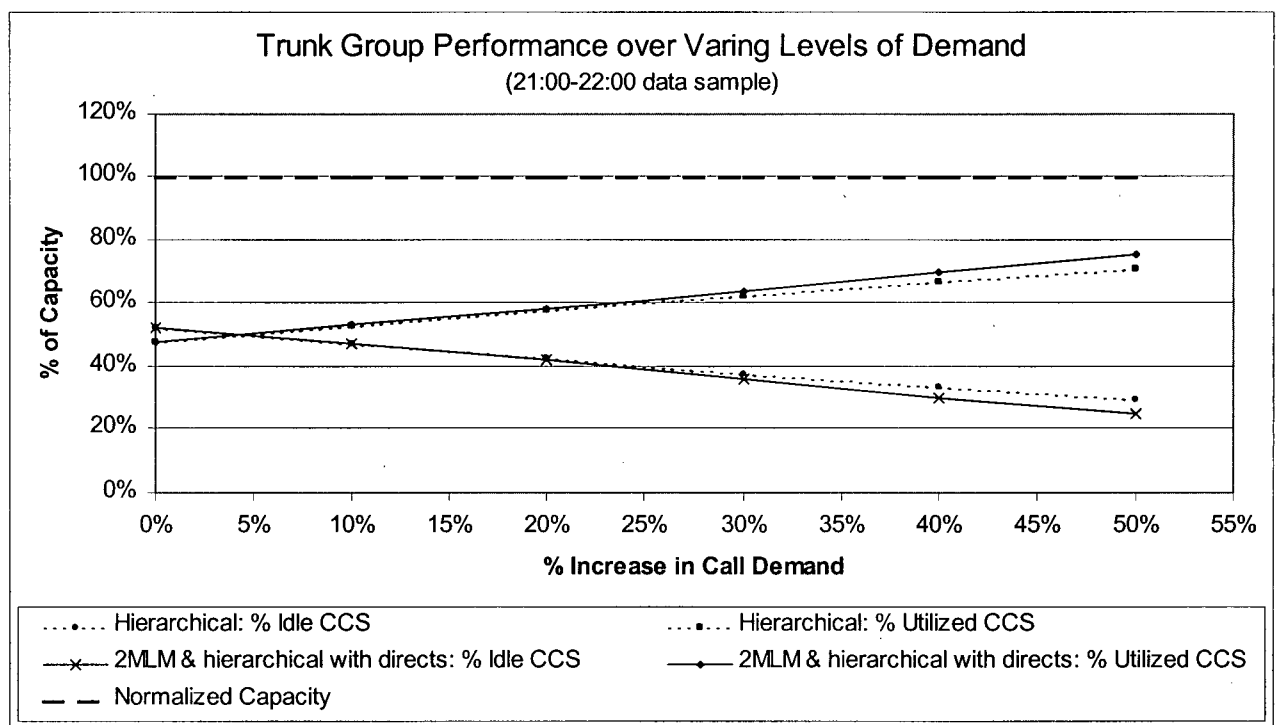
Figure 21 – The effect of increasing call attempts upon utilized and idle CCS for the 21:00-22:00 time interval

Sensitivity of Performance to Trunk Reservation Decisions

Two questions were of particular interest. First, what is the effect of applying varying uniform trunk reservation levels for direct trunk groups upon network performance? Second, if trunk reservation was implemented without changes to the routing rules (i.e., with hierarchical routing), what would the effect be upon network performance?

In order to assess the sensitivity of the percentage of blocked calls in the network to different trunk reservation levels, the percentage of trunks reserved on each link was varied.[25] Figure 22 presents the results from varying the percentage of trunks reserved between 0% and 10% for a 40% increase in call attempts during the 11:00-12:00 time interval. There appeared to be a slight decrease in percent blockage as the trunk reservation parameter increased from 0% to 8%. However, the mean percentage of blocked calls for each scenario was within the 99% confidence intervals of the other scenarios. Hence, the percentage of blocked calls in the network appears to be quite insensitive to the percentage of trunks reserved.

---

[25] In cases where applying the percent reservation to trunk groups resulted in a fractional number of trunks, we rounded to the nearest whole number.

% Blocked Vs. % Reserved Trunks
(40% Increase in Call Demand: 11:00-12:00 data sample)
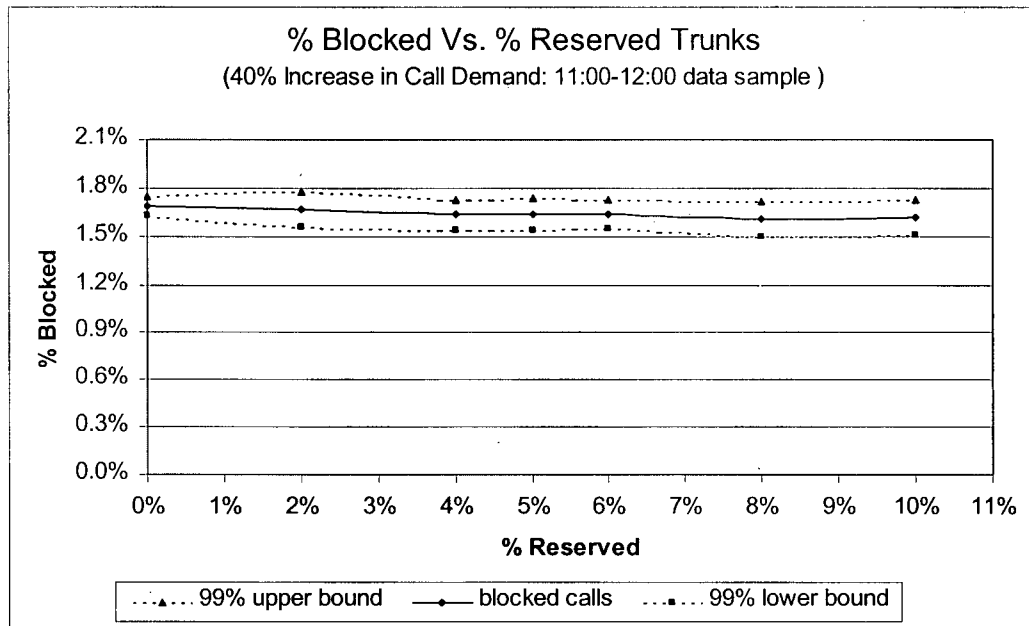
Figure 22 – Percent calls blocked graphed over reservation parameters from 0-10%

To address the second question, two simulations were run with a 50% increase in call attempts and 5% trunk reservation. One with the hierarchical routing rules on the 11:00-12:00 data sample and one with the hierarchical routing rules on the 21:00-22:00 data sample. Relatively small improvements to the percentage of blocked calls were achieved in both cases (Figures 23 and 24), which implies that implementation of a uniform trunk reservation policy alone is not worthwhile.
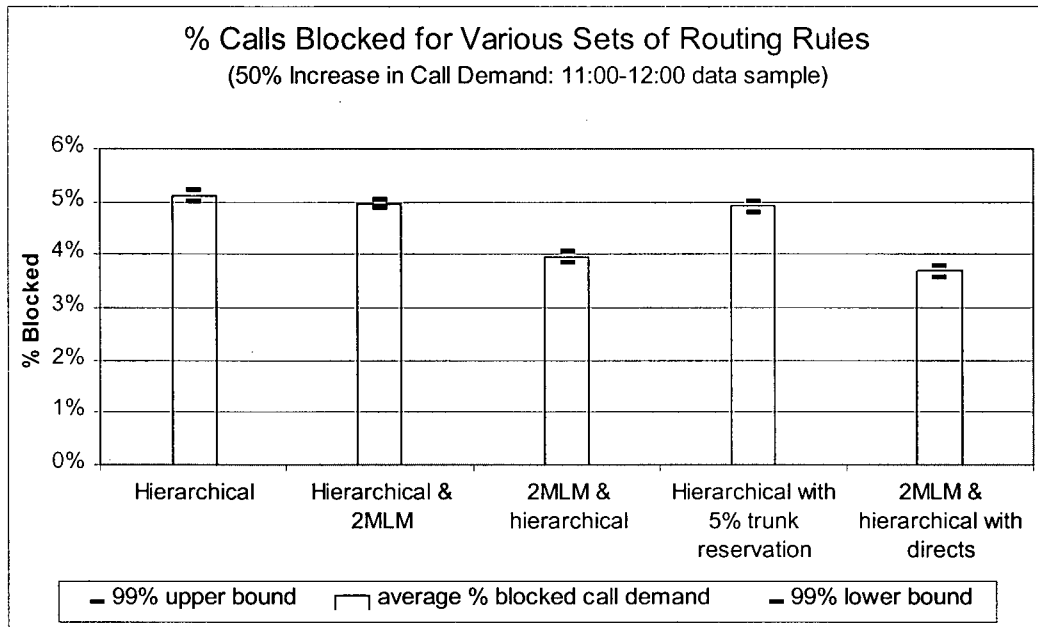
Figure 23 – The relative performance of hierarchical routing with 5% trunk reservation
for the 11:00-12:00 time interval[26]
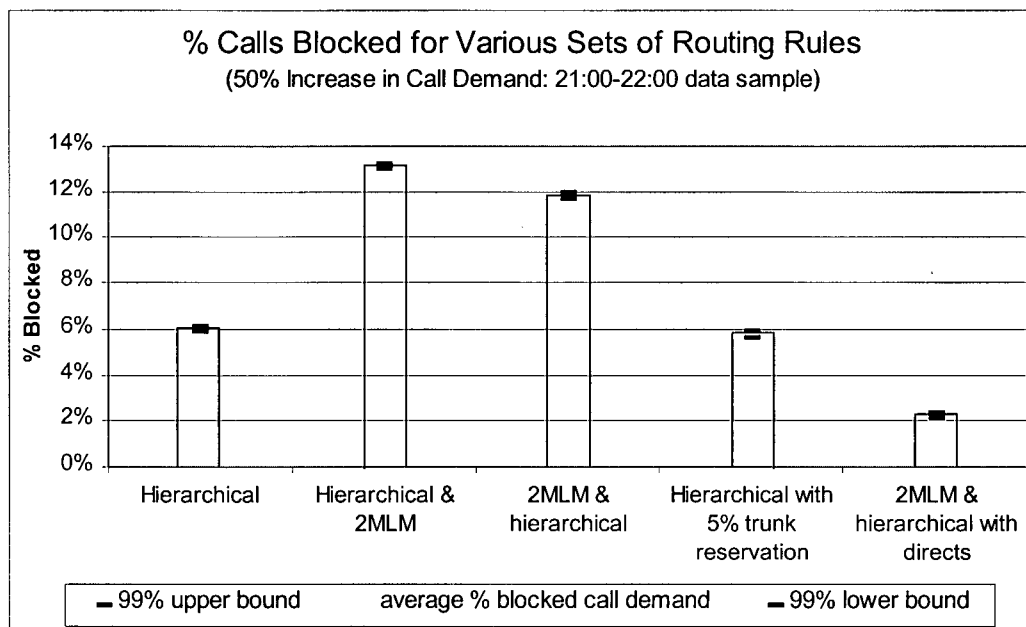
Figure 24 – The relative performance of hierarchical routing with 5% trunk reservation
for the 21:00-22:00 time interval

---

[26] Results from other scenarios are presented again for a relative basis of comparison.

Time of Day Routing

The best performing routes for the 11:00-12:00 and the 21:00-22:00 intervals were different. This would lead one to believe that time of day routing was indeed beneficial. Before developing metrics to quantify and present the differences between the routing rules, we decided to test the best set from each time interval on a simulation of the other time interval. If a set of routes from the 11:00-12:00 interval performed reasonably well in the 21:00-22:00 time interval (or vice-versa), then time of day routing would not be necessary.

Figure 25 presents the relative performance of the best set of routing rules from the 21:00-22:00 interval on the 11:00-12:00 interval. The performance of the rules is almost as good as the ones optimized for the 11:00-12:00 interval. Hence, they could be used with a realized gain of 1% at a 50% increase in call attempts.
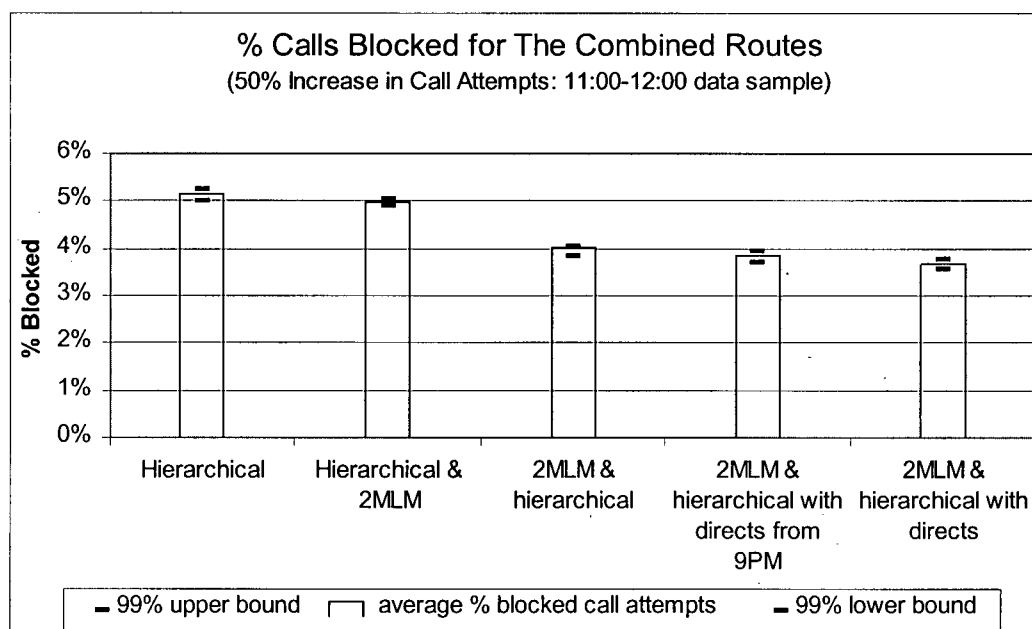
Figure 25 – The relative performance of the 21:00-22:00 *2MLM & hierarchical with directs* routing rules on the 11:00-12:00 data sample

Figure 26 presents the relative performance of the best set of routing rules from the 11:00-12:00 time interval on the 21:00-22:00 time slot. They outperform the rules optimized for the 21:00-22:00 interval.
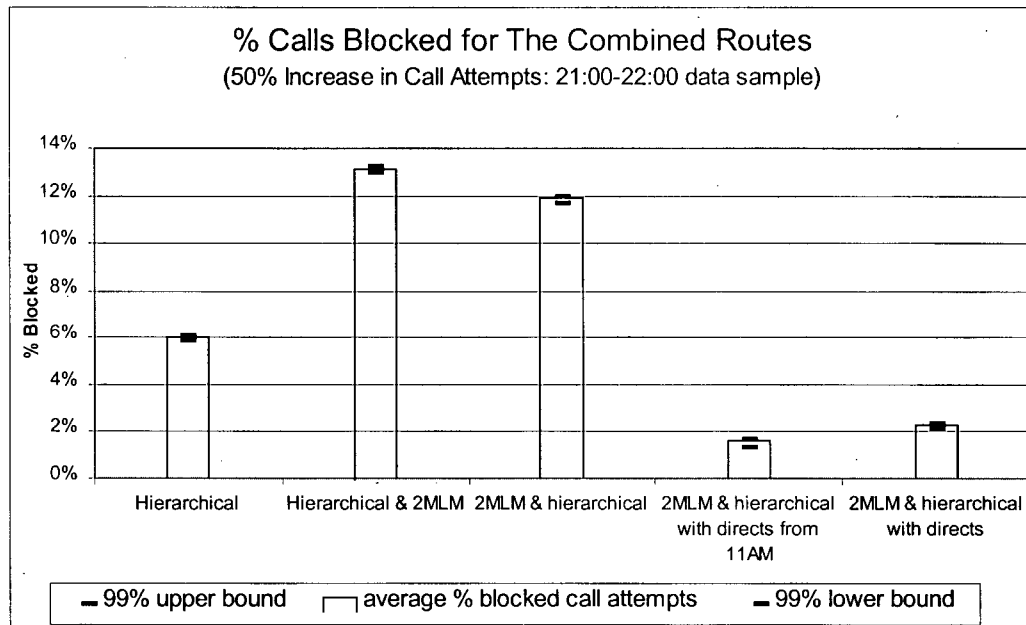
Figure 26 – The relative performance of the 11:00-12:00 *2MLM & hierarchical with directs* routing rules on the 21:00-22:00 data sample

One hypothesis for the performance improvement was the presence of localized suggestion in the 21:00-22:00 time slot. The 2MLM minimizes the maximum average number of trunks required on each. Thus, if one area of the network was extremely strained, the 2MLM would focus its attention there and perhaps not do quite as good a job routing traffic in other areas. Table 9 shows the top 10 links for total number of calls blocked for the 11:00-12:00 scenario and the 21:00-22:00 scenario.[27] The hypothesis was correct. 65 of the 992 OD-pairs in the 21:00-22:00 scenario were blocking at 100%, which heavily contribute to the overall total percentage of blocked calls.

---

[27] The routing rules that produced the lowest percentage of blocked calls for each of the two scenarios was used.

Switches		11:00-12:00		Switches		21:00-22:00	
Origin	Dest.	Calls	% Direct	Origin	Dest.	Calls	% Direct
11	2	3,284	40%	31	18	9,691	100%
22	2	1,960	25%	30	18	7,751	100%
6	10	1,925	13%	30	2	4,651	100%
4	3	1,801	17%	30	23	4,417	100%
2	11	1,780	40%	30	4	4,046	100%
3	4	1,661	17%	30	24	3,920	100%
13	2	1,623	19%	31	4	3,803	100%
4	13	1,583	13%	31	2	3,795	100%
15	2	1,495	22%	30	22	3,196	100%
6	2	1,438	25%	31	22	2,880	100%

Table 9 – 10 highest blocking trunk groups for 11:00-12:00 and 21:00-22:00 intervals utilizing best routing rules

VII.   CONCLUSION

Based upon the results presented in chapter VI, it appears that TELUS could benefit from implementation of alternate nonhierarchical routing rules.   Specifically, employing the combined routing rules – that attempt the direct route first, then attempt the 2-link Maximum Linear Model routes from the 11:00-12:00 time interval, and finally the currently used alternate routes – generated the best routing table of those tested.   Furthermore, these routing rules can be employed throughout the day without the need for change. Logistically, this makes implementation much easier for TELUS because the crossover to new routing rules in each switch does not have to be completely synchronized with other switches on a daily basis.

The results from the trunk reservation analysis suggest that implementation of a uniform policy would be of marginal value. Sensitivity of the *average percentage of blocked calls* in the network to changes in the uniform trunk reservation parameter appears to be minimal.   In other words, the gains achieved from implementation of new routing rules appear to be insensitive to the percentage of trunks reserved on each link.   Finally, compared to Akinpelu's and Krupp's findings of instabilities in nonhierarchical networks without trunk reservation (see chapter II), no instabilities were found for this particular implementation of trunk reservation in the TELUS network.   However, extensive analysis was not performed to ensure that instabilities would not occur.

Consequences of Alternate Nonhierarchical Routing on Dimensioning and Monitoring

As mentioned in the introduction, a move to alternate nonhierarchical routing (ANHR) in the network will affect dimensioning and monitoring. Current dimensioning processes are designed for an alternate hierarchical network. If TELUS decides to move away from this network design, the dimensioning policies will most likely require change. Consequently a new knowledge base for dimensioning a network employing ANHR will have to be built within TELUS. Furthermore, more judicious monitoring of an alternate nonhierarchical network is required. For example, each switch is programmed with instructions for routing. If one switch was programmed incorrectly, cycling could occur in the network[28] causing the number of blocked calls to skyrocket.

Future Research and Development

These results are presented with cautious optimism. First, the suggested routing rules from any optimization method will be highly dependent on the non-probabilistic proxy for demand that they use as input. Consequently, this thesis was a proof of concept. A more comprehensive data sample should be collected over several days to ensure a representative network activity sample is obtained. Specifically, the hours with the highest percentage of blocked calls should be used in the analysis.

Routing and Trunk Reservation

A uniform trunk reservation policy was utilized for the analysis in this thesis and proved to be of little value. However, a non-uniform policy was not explored, which could possibly have significant potential. Furthermore, more rigorous tests for network instabilities without trunk reservation should be performed. Lastly, the interaction between optimal routing rules and optimal trunk reservation levels should be explored. One possibility is the development of an iterative search heuristic to find a good routing/reservation combination.

---

[28] A call "cycles" if it never finds its intended destination switch. Instead, it continues to consume circuits until none are available.

Runtime of The Simulation

The runtime of the simulation limited the number of scenarios that could feasibly be tested. Work is currently underway to decrease the run time. Currently, the simulation generates call attempts and call duration between each OD-pair from empirical distributions. A move to generating call demand and call duration from theoretical distributions would greatly decrease the run time.[29] If this can be achieved without compromising the integrity of the simulation, it will be incorporated in the near future.

Non-Stationary Simulation

One limiting assumption of the analysis was the assumption that the underlying process was in fact stationary. The behavior of demand placed upon the network is in fact non-stationary. Development of a non-stationary simulator in the future would more accurately represent the network's behaviour. Furthermore, the non-stationary simulator could be started in the early morning of a simulated day when carried traffic is almost negligible, thus circumventing the need for a warm-up period.

Large Deviation Theory Applied to Call Blockage

If the demand place upon the network had not been scaled up, the confidence intervals around each solution would not be tight enough to assess the relative performance of the better sets of routing rules on the simulated network. Large deviation theory could be used to reduce the variance associated with the rare event of call blockage. Although the theory is based upon simple ideas, a practitioner should note that the mathematical details to invoke such analysis can sometimes prove quite unwieldy.

---

[29] The simulation would not have to search through discrete bins to find the generated value from the empirical distribution. Instead it would sample from a theoretical distribution with one call to a function.

APPENDICES

Appendix 1 – A sample of the average number of lines required between all OD-pairs

The table below shows the average number of lines required between all OD-pairs, with Switch i as columns (01–31) and Switch j as rows (01–31).

j \ i	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
01	0.0	33.9	32.4	323.6	21.7	45.0	14.7	49.0	24.7	26.2	51.6	26.4	58.7	23.7	51.0	7.2	3.6	36.1	11.9	13.4	42.0	33.5	17.8	18.0	16.7	11.5	5.1	9.1	8.9	26.3	39.5
02	41.3	0.0	85.7	212.7	42.0	107.8	24.7	97.4	51.3	62.8	138.5	65.8	136.5	50.7	109.2	20.0	8.9	78.6	27.4	25.7	95.2	107.0	56.7	57.3	42.2	52.5	49.8	41.0	20.1	80.6	85.9
03	27.6	82.8	0.0	275.1	29.2	41.4	32.1	54.8	23.7	25.4	54.4	29.1	77.6	16.5	48.5	6.5	5.3	26.7	13.0	9.9	34.8	33.6	16.2	13.1	10.7	3.9	5.1	9.2	23.4	27.5	
04	90.1	229.9	248.4	0.0	139.8	250.1	63.5	249.2	129.9	111.0	243.6	144.5	282.7	117.5	257.3	35.0	25.1	127.0	65.7	55.0	156.4	120.2	115.8	111.1	95.3	68.6	20.1	42.7	53.6	171.6	175.5
05	10.0	32.5	16.0	60.1	0.0	102.6	0.0	0.0	0.0	1.8	29.6	11.0	11.6	4.6	18.2	1.8	1.2	0.0	3.7	27.8	17.5	2.8	2.5	4.4	1.9	1.2	1.4	2.2	3.9	8.6	
06	25.2	68.8	38.5	132.6	170.7	0.0	11.9	44.2	18.0	31.0	114.9	24.0	69.9	13.9	52.6	5.4	4.4	48.6	16.5	13.6	56.5	46.7	12.7	10.8	8.4	9.0	5.3	6.9	5.0	19.2	31.0
07	7.4	21.3	16.0	51.8	0.0	15.7	0.0	0.0	0.0	0.0	44.1	15.3	19.0	13.1	19.9	2.7	3.3	0.0	5.0	14.0	12.6	7.5	7.0	8.3	2.1	1.0	3.8	2.2	8.4	9.1	
08	31.2	72.7	59.3	196.6	0.0	58.9	0.0	0.0	0.0	0.0	121.2	77.1	55.9	29.2	65.9	9.5	8.8	0.0	0.0	48.3	45.2	21.2	19.0	21.4	12.0	6.2	7.3	7.2	23.0	29.8	
09	14.2	36.4	16.1	78.1	0.0	21.0	58.9	0.0	0.0	0.0	45.5	37.6	33.5	21.1	34.1	6.2	4.1	0.0	0.4	3.8	25.6	23.4	11.1	9.0	10.3	6.7	5.8	3.3	4.9	11.2	11.5
10	21.6	70.4	33.0	113.8	3.6	128.6	0.0	0.0	6.9	0.0	147.0	60.0	38.1	17.1	70.7	7.0	3.2	0.0	0.3	10.4	35.8	41.0	14.4	12.1	41.9	15.8	9.5	7.9	21.5	25.9	
11	37.7	79.2	50.5	175.1	32.0	82.4	34.7	136.1	52.4	80.1	0.0	98.5	82.7	52.9	109.8	8.6	26.1	51.9	15.0	19.8	78.7	75.6	43.7	24.1	46.5	11.4	23.5	25.7	18.4	35.1	75.3
12	17.9	42.0	18.2	92.9	11.5	21.1	14.3	43.8	45.0	28.9	88.7	0.0	49.0	25.8	79.8	8.9	3.6	31.5	8.1	8.8	31.9	43.6	25.7	26.6	7.9	8.7	5.5	6.5	7.4	28.6	18.1
13	39.7	75.3	40.8	274.5	16.6	47.8	20.0	61.1	50.1	61.2	96.6	66.2	0.0	222.3	88.0	50.8	28.1	57.7	19.0	12.1	72.5	86.9	44.2	42.0	36.6	10.5	14.8	12.4	48.0	41.8	
14	15.1	25.7	10.7	68.3	6.0	14.1	10.0	22.6	23.8	9.3	31.9	25.9	170.7	0.0	28.4	11.3	8.1	17.7	7.7	4.6	26.9	36.3	14.9	15.4	55.6	7.5	3.7	5.7	3.3	15.7	11.2
15	36.3	88.5	42.1	210.2	27.3	68.7	57.6	71.2	46.9	75.6	121.4	99.2	85.1	38.4	0.0	35.5	29.8	121.6	78.7	35.7	171.9	123.0	46.4	46.1	27.9	25.7	13.1	23.4	27.3	78.2	88.1
16	5.5	10.1	5.7	32.0	2.7	6.6	2.6	7.2	4.4	4.3	11.8	8.1	28.7	16.8	0.0	11.5	0.0	23.5	7.4	3.6	3.0	32.1	18.0	13.6	16.3	9.0	6.5	4.2	4.0	2.5	12.9
17	2.5	5.6	4.6	12.9	1.6	4.5	2.9	4.7	5.6	1.5	6.9	3.7	10.5	5.9	6.4	23.2	0.0	3.2	1.5	1.1	13.2	7.6	4.1	3.9	3.1	1.1	1.6	0.4	5.2	2.2	
18	34.2	69.4	56.3	148.9	0.0	80.8	0.0	0.0	0.0	0.0	78.7	47.8	108.4	37.1	163.5	26.7	21.0	0.0	63.1	80.1	114.0	77.9	33.2	26.9	29.1	11.6	22.9	49.8	103.7	158.2	
19	12.6	24.8	14.9	95.9	0.0	24.3	0.0	0.0	0.0	0.0	23.8	15.7	29.0	10.2	58.4	4.6	2.2	0.0	26.8	73.0	46.9	18.0	12.2	9.9	10.2	3.4	7.2	10.6	22.2	40.5	
20	8.3	13.5	6.3	21.5	4.1	11.8	2.2	8.3	4.9	8.6	15.8	7.6	16.7	3.0	29.7	1.3	0.7	22.1	22.5	0.0	34.5	37.5	7.1	6.0	3.3	5.0	2.7	4.9	14.2	16.6	80.1
21	26.2	50.8	29.0	143.5	12.8	46.9	31.3	134.6	53.6	22.4	70.0	32.9	86.0	23.5	138.0	15.0	8.4	59.6	54.4	52.8	0.0	183.4	46.0	73.7	22.9	11.6	36.2	89.1	65.4	71.3	
22	29.2	64.4	30.0	123.7	20.5	53.1	11.2	40.3	23.7	26.1	72.0	42.5	96.8	32.0	100.6	13.5	7.2	55.3	30.9	38.1	160.7	0.0	49.7	49.3	24.4	15.0	23.1	25.3	70.4	85.6	
23	12.0	26.5	8.2	55.4	3.6	14.9	4.6	11.9	8.7	10.1	27.4	25.5	38.7	18.3	47.6	16.3	3.8	19.0	9.0	8.4	72.3	57.4	0.0	166.3	44.4	63.3	15.9	24.1	4.9	148.2	25.6
24	11.8	27.2	11.5	55.3	3.8	10.1	2.6	12.5	8.2	6.9	28.9	24.0	39.5	17.1	40.7	18.1	5.2	22.1	17.4	5.2	58.3	54.0	193.7	0.0	35.7	27.5	25.5	17.6	5.2	142.9	20.2
25	11.2	17.4	11.2	56.4	4.5	12.7	4.0	16.9	7.9	5.3	19.3	9.3	24.4	13.0	24.1	7.0	7.4	13.3	4.8	4.9	45.6	30.3	42.1	36.8	0.0	33.4	11.9	14.4	4.1	37.1	13.0
26	10.7	17.0	7.2	52.3	4.2	8.5	1.4	8.5	5.6	6.8	18.8	10.8	21.3	6.4	32.1	5.7	2.3	17.3	7.8	6.2	54.8	40.8	66.8	37.3	43.4	0.0	55.9	65.9	9.0	52.3	17.3
27	2.6	11.7	3.3	27.2	1.5	4.4	1.4	4.7	2.6	2.2	6.5	5.7	10.1	3.1	17.0	2.8	1.1	6.3	3.5	4.5	19.0	17.4	16.5	21.7	12.3	48.9	0.0	18.8	5.3	21.5	9.3
28	6.2	17.6	6.3	38.9	2.6	8.1	0.9	5.1	2.6	5.6	11.9	8.7	36.3	5.0	18.8	3.7	1.7	17.1	5.7	3.4	29.4	27.2	24.9	16.6	13.4	51.2	0.0	0.0	23.1	14.5	96.7
29	5.4	11.3	4.0	22.6	3.1	8.5	1.0	6.1	1.6	3.1	11.8	6.5	7.3	2.1	18.2	1.0	0.8	14.2	5.2	13.1	31.0	32.2	8.0	4.2	4.2	5.4	3.8	5.2	0.0	10.7	96.7
30	12.8	38.3	16.6	78.3	4.1	21.7	5.7	17.4	11.7	16.9	39.3	27.1	39.4	23.9	64.3	15.8	4.4	32.6	15.7	53.3	89.4	84.5	171.2	160.0	56.5	101.9	30.9	81.5	14.6	0.0	48.3
31	23.4	41.2	16.1	81.1	9.4	31.6	5.6	21.6	9.5	20.0	48.2	24.5	57.2	10.4	65.5	5.0	3.8	54.4	32.7	83.8	92.1	104.5	22.5	18.8	13.3	7.5	12.8	96.9	41.4	48.3	0.0

Appendix 2 – Sample of the output from the MFCM

Origin Dest.	1 2	1 3	1 4	1 5	1 6	1 7	1 8	1 9	32 27	32 28	32 29	32 30	32 31
Route 1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	32,27	32,28	32,29	32,30	32,31
Route 2									32,16,27	32,23,7,17,28	32,11,29	32,23,7,17,30	32,30,31
Route 3									32,23,7,17,27	32,2,17,28	32,30,11,29	32,2,17,30	32,23,6,30,31
Route 4									32,2,17,27	32,23,6,17,28	32,23,6,30,11,29	32,23,6,17,30	32,23,7,30,31
Route 5									32,23,6,17,27	32,14,2,17,28	32,23,7,30,11,29	32,14,2,17,30	32,23,7,17,30,31
Route 6									32,14,2,17,27	32,12,7,17,28	32,23,7,17,30,11,29	32,12,7,17,30	32,23,7,18,30,31
Route 7									32,12,7,17,27	32,16,7,17,28	32,23,7,18,30,11,29	32,16,7,17,30	32,21,24,30,31
Route 8									32,16,7,17,27	32,23,7,18,28	32,21,24,30,11,29	32,23,7,18,30	32,21,25,30,31
Route 9									32,23,7,18,27	32,2,18,28	32,21,25,30,11,29	32,2,18,30	32,21,26,30,31
Route 10									32,2,18,27	32,23,6,18,28	32,21,26,30,11,29	32,23,6,18,30	32,21,27,30,31

REFERENCES

[1] Bertsimas, Dimitris, Tsitsiklis, John, *Introduction to Linear Optimization*, Athena Scientific, Boston Massachusetts (1997).

[2] Girard, André, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley (1990) 274-277.

[3] Goto, Jason et al., Centre for Operations Excellence, University of British Columbia (2000).

[4] Jagerman, D. L. (1984), Methods in Traffic Calculations, *AT&T Bell Laboratories Technical Journal,* Vol.63, No.7, September 1984 1283-1310.

[5] Queyranne, Maurice, professor of Management Science, University of British Columbia (2000).

[6] Smith, Isabelle, *An Application of Multivariate Analysis to Time of Day Routing in Telecommunication Networks*, Masters Thesis, Centre for Operations Excellence, University of British Columbia (2000).