1

ABSTRACT

This document describes a method for reducing the required number of circuits or transmission channels when dimensioning a telephone voice network for multiple time periods of varying traffic load, at a given service level. The proposed method accomplishes the circuit reduction by rerouting traffic in a new time period to utilize spare circuit capacity established in previous periods. Sample savings of approximately one percent circuit reduction have been found for the application of the method to two test networks representative of existing networks. Introductory material on telephone traffic and network dimensioning has also been included.

0-315-19913-X

National Library
of CanadaBibliothèque nationale
du Canada

Canadian Theses Division Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

67825

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Ecrire en lettres moulees ou dactylographier

Full Name of Author — Nom complet de l'auteur

GEORGE YEE

Date of Birth — Date de naissance

MAY 5, 1949

Country of Birth — Lieu de naissance

CHINA

Permanent Address — Résidence fixe

244 OWL DRIVE,
OTTAWA, ONT.
K1V 9J6

Title of Thesis — Titre de la thèse

OPTIMAL ROUTING IN DIMENSIONING COMMUNICATION
NETWORKS WITH TIME-VARYING TRAFFIC DEMAND

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M.Sc.

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

PROF. E. J. NORMINTON

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies of
the filmThe author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permissionL'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et de
prêter ou de vendre des exemplaires du filmL'auteur se réserve les autres droits de publication, ni la thèse
ni de longs extraits de celle-ci ne doivent être imprimés ou
autrement reproduits sans l'autorisation écrite de l'auteur

Date

April 3, 1985

Signature

George Yee

OPTIMAL ROUTING IN DIMENSIONING COMMUNICATION
NETWORKS WITH TIME-VARYING TRAFFIC DEMAND

by

George Yee, B.Sc.,

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Science,

Department of Mathematics and Statistics,
Carleton University,
Ottawa, Ontario,

March 1, 1985.

© copywrite
1985, George Yee

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

"Optimal Routing in Dimensioning Communication
Networks With Time-Varying Traffic Demand"

submitted by George Yee, B.Sc.,
in partial fulfillment of the requirements for
the degree of Master of Science.

Thesis Supervisor

Chairman, Department of Mathematics and Statistics,

Carleton University,

Date: _____

ABSTRACT

This document describes a method for reducing the required number of circuits or transmission channels when dimensioning a telephone voice network for multiple time periods of varying traffic load, at a given service level. The proposed method accomplishes the circuit reduction by rerouting traffic in a new time period to utilize spare circuit capacity established in previous periods. Sample savings of approximately one percent circuit reduction have been found for the application of the method to two test networks representative of existing networks. Introductory material on telephone traffic and network dimensioning has also been included.

ACKNOWLEDGEMENTS

I wish to thank T.M. Yuen, Assistant Vice-President, Networks Technology, Bell-Northern Research, for granting me a sabbatical from employment in which to research this work. I am very grateful to my thesis Supervisor, Professor E.J. Norminton of Carleton University, for his support and encouragement throughout the many delays in the submission of this thesis. I am also indebted to Dr. M.E. Beshai and L.E.N. Delbrouck of Bell-Northern Research for their support and suggestions throughout this exercise.

Finally, I wish to thank my wife Corinne for her patience and understanding during those times when I totally shut myself out from family life to work at this endeavour.

TABLE OF CONTENTS

ACCEPTANCE SHEET	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. TELEPHONE TRAFFIC	2
2.1 Telephone Traffic Defined	2
2.2 Units of Telephone Traffic	5
2.3 The Behaviour of Telephone Traffic	6
2.4 Linking Subscribers Together	14
2.4.1 A First Naive Approach	15
2.4.2 Switching Introduced	16
2.5 An Optimization Problem	17
2.6 Teletraffic Engineering	20
2.6.1 The Busy Hour	21
2.6.2 Network Trunk Sizing	22
2.6.3 Multi-hour Engineering	24
2.7 Summary	28
CHAPTER 3. NETWORK SIZING	29
3.1 Telephone Traffic and Probability Theory	29
3.1.1 Smooth, Rough, and Random Traffic	30
3.1.2 Traffic Sources and the Rate of Call Arrival	32
3.1.3 Limited Vs. Full Access	33
3.1.4 Treatment of Lost Calls	33
3.1.5 The Erlang-B Formula	34

3.1.5.1	Derivation of the Erlang-B Formula . .	36
3.2	Trunk Group Efficiency	41
3.3	Trunk Efficiency and Alternate Routing	47
3.4	The North American Switching Hierarchy	49
3.4.1	Call Routing	54
3.5	Elements Of Network Sizing	57
3.5.1	The Marginal Load (ML) Method	57
3.5.2	Equivalent Random Method (ERM)	62
3.5.2.1	Using ERM in ML . .	65
3.6	Mechanics of Network Sizing	68
3.6.1	A Sizing Algorithm	68
3.6.2	Problem Complexity	70
3.7	Summary	71
CHAPTER 4.	MULTI-HOUR ENGINEERING	72
4.1	Selection of Traffic Matrices	72
4.2	Internal and External Point-to-Point Traffic	73
4.3	Sizing Up	74
4.3.1	Rationale of Sizing Up	77
4.3.2	Drawback of Sizing Up	77
4.4	Improving the Sizing Up Method	78
4.4.1	The Extended Sizing Up Method (ESU)	79
4.4.1.1	Traffic Decoupled Sets	79
4.4.1.2	Extended Sizing Up - Idealized Case	82
4.4.1.3	Extended Sizing Up -Algorithm	84
4.4.2	Sizing Up With Rerouting (SUR)	87
4.5	Summary	88
CHAPTER 5.	SIZING UP WITH REROUTING (SUR)	89
5.1	Temporal Traffic Variations and Trunk Occupancy	89

5.2	Utilization of Idle Trunk Capacity Through Rerouting	92
5.2.1	Criteria for Rerouting	93
5.2.2	Approximate Criteria for Rerouting	94
5.3	The Rerouting Algorithm	96
5.3.1	Routing Loops	102
5.3.2	Analysis of the Rerouting Algorithm	103
5.4	Sizing Up With Rerouting: Algorithm	109
5.4.1	Explanation of the SUR Algorithm	113
5.5	Summary	114
CHAPTER 6.	SIZING UP WITH REROUTING: EXPERIMENTAL RESULTS	116
6.1	Test Networks	116
6.2	Computer Runs	120
6.3	Experimental Results	121
6.3.1	Network 34 Results	122
6.3.2	Network 152 Results	123
6.4	Distribution of Trunks: SU vs. SUR	124
6.5	Summary	126
CHAPTER 7.	SIZING UP WITH REROUTING: CONCLUSIONS	127
7.1	Conclusions on SUR	128
7.2	Future Work	129
7.3	Summary	132
REFERENCES		133

LIST OF FIGURES

Figure 1.	Seasonal Variations in a Typical 2000-Line City Exchange	8
Figure 2.	Typical Daily Traffic Variation	10
Figure 3.	Distribution of Traffic During a Typical 24-Hour Period	12
Figure 4.	Variation of Traffic On a 30-Line Trunk Group Sampled Every Minute for One Hour .	13
Figure 4A.	Typical Traffic Intensity Variation	26
Figure 4B.	Hourly Traffic Load Variation	27
Figure 5.	The Loss System M/M/n/n/oo	37
Figure 6A.	Occupancy of a Trunk Group at 5% Blocking	43
Figure 6B.	Occupancy of a Trunk Group Offered 120 Erlangs of Traffic	44
Figure 7.	Blocking Vs. Occupancy For a Trunk Group of 50 Trunks	46
Figure 8.	The North American Switching Hierarchy .	53
Figure 9.	Sample 7-Node Routing	56
Figure 10.	Cost Components of the ML Method	61
Figure 11.	Components of the Equivalent Random Method	64
Figure 12.	Applying ML and ERM to Size IHU Groups .	67
Figure 13.	Sizing Up	76
Figure 14.	Traffic Decoupled Subsets of Node Pairs	81
Figure 15.	Extended Sizing Up	86
Figure 16.	Effects of Internal Traffic Variations in SU	91
Figure 17.	Flowchart of the Rerouting Algorithm . .	101
Figure 18.	The Three Tandem Types in the Rerouting Algorithm	105
Figure 19.	Flowchart of the SUR Algorithm	112

Figure 20. Network 34 Nodes Distribution	117
Figure 21. Network 152 Nodes Distribution	119

CHAPTER 1. INTRODUCTION

The material presented here has been organized into two parts. The first part contains background information. The second part deals with the objective of this work: the presentation of a new methodology to reduce the number of circuits required for a communication network subjected to temporal variations in traffic load. The first part is made up of chapters 2 to 4. The second part is comprised of chapters 5 to 7.

Chapter 2 is an introduction to telephone traffic and its characteristics. Chapter 3 discusses how a telephone network is provisioned for the number of required circuits, based on the traffic load and other criteria. Chapter 4 introduces the idea of temporal variations in traffic load and describes conventional techniques for dimensioning a network to accomodate the varying load. Chapter 5 presents the method mentioned above, for reducing the number of required circuits needed to handle a varying traffic load. Chapter 6 gives the results of applying this method to two test networks. Finally, chapter 7 offers some conclusions on this circuit reduction technique and gives some suggestions for future work.

CHAPTER 2. TELEPHONE TRAFFIC

2.1 Telephone Traffic Defined

The telephone is a device that has extended man's communication capabilities to the point that he can merely pick it up, dial a few digits, and talk to someone half way around the world. The act of talking on the telephone, however, has an effect that is often unnoticed by the telephone subscriber, namely that of generating traffic.

Most of us are familiar with the term "traffic" from first hand experience at commuting between our homes and places of employment. We are all too cognizant of the phrases "traffic jam" and "congestion". The notion of telephone traffic is not too unlike the idea of street vehicular traffic. We also speak of congestion in telephone traffic, for here too there is contention for available resources. In the case of vehicular traffic, congestion arises from contention for highway paths between home and office. In the case of telephone traffic, congestion occurs from contention for wire paths between caller and callee. We shall examine telephone traffic congestion in more detail later, after first exploring some more basic aspects of telephone traffic.

Telephone traffic is, in essence, the sum total of the work done by a group of circuits or trunks during one hour, or the sum total of the telephone calls offered to it during that hour. There are two quantities implicit in this definition, namely the number of calls and the duration of the calls. Traffic flow through a group of trunks during a period of time is measured as the product of the number of calls during that time and the average length of the calls or average holding time. The period of time for this measurement is one hour by convention. This length of time was chosen basically for two reasons: 1) one hour is sufficiently long for accurate measurement, and ii) one hour is sufficiently short to justify the stationary treatment of traffic. By ii) we mean that one hour is just short enough so that the traffic of a certain hour every business day, say, can be considered stationary. As we shall see later, stationarity of traffic is both a requirement and a limitation of current traffic theory. Traffic flow can be expressed as:

$$A = C \times T$$

where C is the average arrival rate, T is the average holding time, and A is the traffic flow.

For example, suppose 100 calls with an average holding time of 3 minutes are generated during one hour, by subscribers connected to a common trunk group. We then say that the traffic flow through the group is $100 \text{ calls} \times 3 \text{ minutes} = 300 \text{ minute-calls}$. Notice that different factors can generate the same 300 minute-calls, for instance, 10 calls in the hour, each lasting an average of 30 minutes, will also generate 300 minute-calls. In particular, 5 calls during the hour, each lasting 60 minutes, generate the same 300 minute-calls. Since each of these calls lasts 60 minutes or one hour, they can be considered as simultaneous calls during the hour of measurement. The number of calls each lasting one hour (simultaneous calls) during the hour of measurement is termed the traffic intensity. For this example, the traffic intensity is 5 calls. It is numerically equal to the traffic flow expressed in hour-calls ($300 \text{ minute-calls} / 60 \text{ minutes per hour}$ gives 5 hour-calls).

The traffic intensity is used by traffic engineers to determine the number and capacity of transmission and switching equipment required in a telephone network. Other terms often encountered in traffic theory are "offered traffic" and "carried traffic". The latter is the amount of traffic actually carried by a trunk group. The former is the quantity of traffic offered or presented to the trunk

group and is greater than the carried traffic by the amount of lost or blocked traffic on the group, if any.

2.2 Units of Telephone Traffic

Since the traffic intensity is always the average number of simultaneous calls in one hour, it is clear that it can be dimensionless and its meaning still understood. However, there are units assigned to it. The unit of traffic intensity recognized internationally is the "erlang", named after the Danish mathematician A.K. Erlang, founder of the theory of telephone traffic. The erlang is used in classical traffic theory. One erlang represents a server occupied for one hour. In our example above where we obtained 5 hour-calls we could also have called it 5 erlangs (each call occupying one trunk for one hour).

The United States uses a different unit of traffic intensity, namely, "CCS" or "hundred-call-second". The traffic intensity in CCS expresses the sum of the number of busy circuits or simultaneous calls observed once every 100 seconds (36 observations in 1 hour). Since there are 3600 seconds in 1 hour, 1 erlang is the same as 36 CCS and one can readily convert from erlangs to CCS and vice versa.

The erlang unit is more meaningful than the CCS unit. The erlang per trunk represents the trunk occupancy, i.e. the proportion of the hour during which the trunk is in use. Suppose, in the above example, that we have 24 trunks carrying 12 erlangs. The occupancy of the trunks would then be 50%. This means that, on the average, each trunk is occupied for one-half hour.

2.3 The Behaviour of Telephone Traffic

To properly determine or dimension the amount of telephone transmission and switching equipment needed, it is necessary to understand the behaviour of telephone traffic with respect to its distribution in time and space. Telephone traffic varies without uniformity according to user needs from one time period to the next. Traffic varies from hour to hour, day to day, month to month, and season to season. There is also variation from minute to minute in the same hour, as well as variation from subscriber to subscriber.

Figure 1 [1] shows the seasonal variation in a typical 2000-line city exchange during a period of one year. The abscissa represents the weeks of the year and the ordinate shows the highest number of calls that have occurred during one of the 168 hours of the week. If we assume that the average call holding time is about the same over all the 52 weeks, the graph gives a good representation of the seasonal variation of the busiest hour of the week. According to this graph, the busiest season extends from October to December.

Figure 1 – Seasonal Variation in a Typical 2000 Line City Exchange

The decision on how much telephone exchange equipment is required is usually based on the busy season. However, there are occasional expected and unexpected traffic peaks. Expected peaks may arise due to such holidays as Mother's Day and Christmas. Unexpected peaks may be due to special events in the community or nationally, unexpected activity of some group of customers, or, some emergency. *It is not practical to provide enough equipment to cover all instances of high volume traffic. As a result, there will be peak periods during which the provisioned equipment will be insufficient to serve the demand and periods when it will be under-utilized.

Weekly and daily variation in traffic level further characterize the traffic busy season. Some weeks may have more traffic than others, possibly due to local conditions affecting the serving office. Daily traffic variation is depicted in figure 2 [1] and forms a fairly consistent pattern within the week, with highs on Mondays and Fridays, lows on Wednesdays, and so on. ▲ On the other hand, this pattern may be greatly perturbed by holidays occurring within the week or other variable influences such as weather.

Figure 2 – Typical Daily Traffic Variation

Traffic variation between the hours of the day is greater than that between any other periods. The ratio of traffic volume of the busiest hour to the least busy hour can be as high as 100:1. The peak business traffic hour can be expected in the late morning. The peak residential traffic hour, on the other hand, can be expected to occur in the evening. The distribution of traffic during a typical 24 hour period is given by figure 3 [1]. As evidenced in this figure, the traffic variation forms a fairly definite pattern. For the purposes of trunk provisioning, only the busiest traffic hours are of primary importance, although network-wide, the traffic volumes of all hours are significant. More will be said on this point later.

Seasonal and hourly variation are largely systematic. We know from experience fairly well what to expect. However, the traffic variation within the hour is not at all systematic. Figure 4 [1] gives the variation of traffic on a 30-line trunk group sampled every minute for one hour. This same graph for another hour might appear quite different, the only similarity being the irregular jagged appearance.

Figure 3 – Distribution of Traffic During a Typical
24 Hour Period

Figure 4 — Variation of Traffic on a 30 Line Trunk Group Sampled Every Minute For One Hour

Factors contributing to the random pattern observed within the hour are firstly, the number of calls and secondly, the call holding time attributable to a particular user. Business users tend to make more calls than residential users. Subscribers of a particular class can exhibit wide variation in calling frequency and conversation time. For local calls, short holding times of one to three minutes are more common than longer holding times of ten or more minutes.

2.4 Linking Subscribers Together

In the above sections we have dealt with traffic patterns and traffic variation. We already have some idea of what is meant by traffic flow on a trunk. To develop these ideas further, we now examine the practical problems of connecting subscribers together to provide a telephone service.

2.4.1 A First Naive Approach

A first approach towards linking subscribers together is to connect them to one another using separate wire paths, lines, or trunks. If this were done, the grade-of-service would be excellent, as no call would ever be blocked through lack of a wire path between subscribers. The situation would resemble the following diagram, drawn for 7 subscribers:

7 subscribers interconnected

However, such a situation would be intolerable for a telephone company. The costs in terms of capital outlay for the separate lines would be prohibitive! For just 7 subscribers, there are already 21 wire paths. The number of wire paths for n subscribers is $n(n-1)/2$. Considering the large number of subscribers, even for a small town, this approach is clearly untenable.

2.4.2 Switching Introduced

The preceding situation can be remedied by the introduction of switching. It is a fact that subscribers on the average use their telephones for only a small fraction of the day. If they were all connected by separate lines, most of the lines would be idle at any given time. We can reduce the number of lines in the previous situation by connecting each subscriber to a switch and then letting the switches connect or switch subscribers together, as in the following situation:

Here the total number of lines is 7. We have added 2 switches and a collection of wire paths between the switches. Such a collection of wire paths is, as we know by now, termed a trunk group, within which each circuit is termed a trunk. As we see, we have reduced the total number

of copper paths at the expense of adding 2 switches and a trunk group.

2.6 An Optimization Problem

In general, as we reduce the number of lines, we increase the number of switches and trunk groups. Switches and trunk groups cost money as well as lines. Therefore, there is an optimal balance to be sought out among quantities of lines, switches, and trunk groups. We have arrived at an optimization problem. However, the situation up to now has been very simplistic. Into the optimization problem must be injected such factors as:

- i) concentration,
- ii) routing and trunk sharing,
- iii) communities of interest.

Concentration describes the number of lines assigned to a particular switch. The higher the number of lines for a particular switch (at a given switch capacity), the higher the concentration (line concentration) at that switch. As we shall see later, larger trunk groups are more efficient at carrying traffic (i.e. the larger the trunk group, the

higher is the ratio of traffic carried on the group to the number of trunks in the group, at fixed blocking - see section 3.2). However, the size of the trunk group may be limited by the line demand in the serving area of the switch, the line handling ability of the switch itself, and failure considerations (the larger the group, the greater the number of people affected should that group fail).

Routing and trunk sharing may be illustrated with the following network:

Here, we have 4 switches connected by 6 trunk groups with suitable concentrations of lines at each switch. Suppose the trunk group connecting switches 1 and 4 is nonexistent. Then traffic originating at switch 1 destined for switch 4 can arrive at switch 4 via 3 different paths or routes, namely 1-3-4, 1-2-4, and 1-2-3-4. Actually, there is a much larger number of routes from switch 1 to switch 4 if we allow path cycles to occur. For instance, the path

1-2-3-1-2-3-4 has the cycle 1-2-3-1. However, we explicitly do not allow cycles since they waste network resources and contribute nothing towards call processing. The availability of a choice of routes, as just given, means that there is a need to establish some rules for routing a call through a network, in order to insure order and consistency. We will describe a set of routing rules in the next chapter.

The fact that most people do not use their telephones all the time and the fact that there are traffic busy hours may be used to reduce the total number of trunks needed in a network. This is done by constructing the routing rules so that they direct busy hour traffic away from congested groups to alternate groups. These alternate groups are thus shared between the traffic normally offered to them (first-offered traffic) and the redirected traffic (overflowed traffic). The congested groups may be carrying busy hour traffic. The alternate group may have trunks that are not being used. If the alternate group is also congested, its traffic may be redirected to yet another alternate group and so on. In the immediately preceding network, suppose traffic from switch 1 to switch 4 finds the 1-4 group congested. Then the routing rules can redirect it to the alternate group 1-3 for completion along the group 3-4. If group 1-3 is congested, the routing rules can redirect the traffic to group 1-2 for completion along group 2-4, and so on. The net result of all this is to make

possible fewer trunks for group 1-4 by sharing trunks on groups 1-3 and 3-4 and possibly also sharing trunks on groups 1-2 and 2-4.

A community of interest between two regions is a measure of the volume of traffic between the two regions. We implicitly assume when we use this term, that traffic volume is directly related to the level of interest that each region has for the other. A high community of interest means a high volume of traffic. Of course, communities of interest impact the way the network is arranged. For example, in the figure directly above, if region 1 (corresponding to switch 1) has a sufficiently high community of interest with region 2 (corresponding to switch 2)*and the 2 regions are not too far apart geographically, we may consider replacing switches 1 and 2 with a single larger switch and doing away with the trunk group 1-2 .

2.6 Teletraffic Engineering

In the preceding sections we have informally introduced some of the concepts in telephone networking - the linking of telephone subscribers together to provide a telephone service. We now proceed on a more formal basis to describe

the issues involved in the actual teletraffic engineering of a telephone network.

The purpose of teletraffic engineering is to economically provision telephone switching and transmission equipment while providing satisfactory customer service at all times. However, in this work, we concern ourselves only with transmission, so that all subsequent references to "provisioning" should be construed as provisioning for transmission paths, or trunks, along which calls are carried between subscribers. When transmission is reduced, switching cost is usually reduced.

2.6.1 The Busy Hour

Telephone equipment provisioning is based on the traffic intensity during the "busy hour" in the busy season.

The busy hour for a network, however, can be defined in several ways. For instance, the busy hour could be the busiest hour of an entire busy season, month, or week. Alternatively, it could be the busiest hour of the busiest day of a busy season, month, or week. Another definition would be the time-consistent busy hour, where the busiest

clock hour, say 10-11 A.M., is averaged over a period of time such as a busy season, month, or week or even the ten high volume days. However, none of these busy hour concepts appear to provide a strong basis for sound engineering in light of the objective of providing satisfactory service at minimal costs. To provision for the busiest single hour of the busy season would lead to idle equipment most of the time. At the other extreme, use of a busy season time-consistent busy hour could mean equipment overload for some clock hour that is busier than the average time-consistent busy hour. Such inadequacies lead to the idea of multi-hour engineering for provisioning a telephone network. Although we will not deal with it here, an alternative to multi-hour engineering is extreme value engineering, in which a network is engineered to a statistically estimated peak traffic load.

2.6.2 Network Trunk Sizing

Network trunk sizing (or simply network sizing) is the process whereby the number of required trunks is determined between switches subject to the traffic demand, economic constraints, and a required service level. The traffic demand is that corresponding to some selected time period, usually some "busy hour" as discussed above. However, a

network sized for only one hour's traffic shows all the inadequacies discussed above and again we refer to multi-hour engineering to be discussed next.

It should be noted that the economic requirements and service level are inversely related to each other. If there were no requirements to keep costs low, subscribers could conceivably be connected to each other by separate trunks or lines as discussed above. This would mean zero call delay or loss: an ideal service indeed. However, no telephone company can operate at unlimited costs. Therefore, subscribers are subject to a service level somewhat less than ideal. In telephony jargon the phrase "service level" is often referred to as "grade of service". The grade of service for a telephone network is usually expressed as the probability of a call being blocked or not getting through due to traffic congestion in the network. A 1% probability of a call being blocked in the network is fairly common. Multiple grade-of-service levels are also used, such as, "1% blocking not to be exceeded 40% of the time, and 10% blocking not to be exceeded 5% of the time".

2.6.3 Multi-hour Engineering

Multi-hour engineering or multi-hour provisioning of a network is an attempt to account for the large variation in traffic intensities discussed above. In this procedure, the time varying traffic is approximated by sets of node-to-node average traffic values obtained by measurement for selected time periods of one hour duration each. Each set of traffic values is usually, for convenience, put in the form of a matrix in which the (i,j) th entry is the amount of traffic from node i to node j . All such traffic matrices are then used for network sizing. The traffic matrices, in effect, represent snapshots of the network at the selected hours. We assume that traffic matrices for a given hour are stationary, i.e. the traffic matrices $\{A(h)\}$ for the same hour h each day, all have the same statistics. (A stochastic process $x(t)$ is stationary if its statistics are not affected by a shift in the time origin. Thus, the two processes $x(t)$ and $x(t+\pi)$ have the same statistics for any π .) Figure 4A is another illustration of the variation in traffic intensity throughout a 24 hour period. The figure also depicts the traffic variation within the hour by jagged lines. The histogram of average hourly traffic can be seen in the figure as a backdrop. Figure 4B presents this histogram by itself. Restating our stationarity assumption, we assume that each column of this histogram is stationary. The alternative treatment of traffic for a given hour as a

nonstationary process is very complex. At the very minimum, it would be necessary to capture and trend the traffic, a task which has not proven practical or economically feasible.

Figure 4A- Typical Traffic Intensity Variation

Figure 4B - Hourly Traffic Load Variation

2.7 Summary

The present chapter has served to introduce the reader to the nature of telephone traffic. Units of traffic have been defined and traffic variation explored. The purpose of teletraffic engineering and some of its techniques have been explained.

In the next chapter we will explore in greater detail the traffic theory involved in sizing a telephone network and present a sizing algorithm.

CHAPTER 3. NETWORK SIZING

In this chapter, we are going to describe how a typical telephone network is sized for the number of required trunks. We will first explore some of the theoretical aspects behind teletraffic engineering, then give some of the rules of network architecture and traffic routing, and end with the mechanics of network sizing.

3.1 Telephone Traffic and Probability Theory

In the last chapter we saw how congestion can arise through contention for limited trunk resources. The probability that a call offered to a trunk group will find an available trunk depends on four factors:

- 1) the distribution of the offered traffic (e.g., random or periodic arrival and constant or exponentially distributed holding time),
- 2) the number of traffic sources- limited or high (infinite),

3) the accessibility of trunks in a group, i.e. full or limited access switches,

4) the manner in which lost calls are treated.

Let us look at each of these four factors in some detail.

3.1.1 Smooth, Rough, and Random Traffic

Probability distributions for telephone traffic fall into three distinct categories depending on the traffic peakedness or variance-to-mean ratio VMR (the ratio of the variance of the distribution to the mean of the distribution). So called "smooth" traffic corresponds to $VMR < 1$. For "rough" traffic, $VMR > 1$. For "random" traffic, $VMR = 1$. (The use of "random" here does not imply that smooth and rough traffic are not random. Of course all traffic is random.) The following diagram illustrates these 3 types of traffic distributions:

By rough traffic we mean traffic that exhibits larger variation than random or smooth traffic. For a given grade of service, more trunks are needed for rough traffic because of more "bursty" arrivals, i.e. the incidence of short inter-arrival times is more pronounced than for random traffic. To see this, consider the following analogy with a city highway. The width of this highway is set to carry a number of cars which may vary from few cars one day to many cars the next day. Now if the number of cars carried is suddenly increased many times, then in order to avoid undue delays, the width of the highway will have to be expanded. Smooth traffic behaves like rough traffic that has been filtered. The filtering is done by the local exchange which sees random traffic coming at it but sends out smoothed traffic due to blockage during the busy hour. Smooth traffic may be described by the positive binomial Bernoulli distribution. This traffic is assumed when dealing with small groups of subscribers. Rough traffic is usually well represented by the negative binomial distribution. Random

traffic is described by the Poisson distribution, which may be derived from the binomial distribution (limiting case as the number of traffic sources approaches infinity).

3.1.2 Traffic Sources and the Rate of Call Arrival

Traffic sources may be assumed finite or infinite. In the infinite regular case (approximated by a very large number of sources), the call arrival rate is independent of the number of calls already in the network. In the peaky traffic cases, one effectively assumes the call arrival rate to grow with the states of the system. In the finite case, the rate of call arrival is proportional to the number of free sources.

In this work we concern ourselves with random traffic from an infinite number of sources, described by the Poisson distribution.

3.1.3 Limited Vs. Full Access

A switch may be considered as a device with inlet and outlet ports. Any inlet may connect to any outlet. With two-way trunking, a port may serve as an inlet or an outlet as desired. A switch is said to have full access when each inlet has access to any outlet. When this is not the case, the switch is said to have limited access. Traffic is blocked in a switch when a given inlet has no access to a particular outlet. In this work, limited access switches will not be considered, as full access switches are more the norm today. In addition, switches will be treated as nonblocking.

3.1.4 Treatment of Lost Calls

There are three methods of handling lost calls in conventional traffic theory: 1) lost calls held (LCH), 2) lost calls cleared (LCC), and 3) lost calls delayed (LCD). LCH means that a call which finds all servers busy will wait for an average period of one service duration. LCC means that the subscriber's call attempt is denied, never to be tried again. LCD means that the subscriber is automatically put in a queue. For example, this occurs whenever the

subscriber is put on hold. Depending on how the queue of waiting calls is handled, LCD has a further three subcategories: the waiting calls may be served last-in first-out, first-in first-out, or at random.

The LCH and LCC categories are further classed as loss systems. The LCD category falls under the heading of delay systems. The telephone circuit-switched voice network is a loss system. Since we concern ourselves in this work with sizing such a network, we are concerned with a loss system. Further, we adopt the LCC method of handling lost calls, as this is the most commonly accepted description of subscriber behaviour.

3.1.5 The Erlang-B Formula

The Erlang-B formula for sizing trunk groups is as follows:

$$B(n, a) = \frac{\frac{a^n}{n!}}{\sum_{i=0}^n \frac{a^i}{i!}}$$

where $B(n, a)$ = the steady state probability of blocking due to congestion,

n = the number of trunks, and

a = the mean of the offered traffic

The Erlang-B formula is based on the assumption that

- . the number of traffic sources is infinite,
- . lost calls are cleared with zero holding time,
- . each calling party has full access to all the trunks in the group under consideration,
- . the system is in the steady state.

The Erlang-B formula is used internationally and has been standardized by the CCITT (International Telegraph and Telephone Consultative Committee) under the auspices of the United Nations.

3.1.5.1 Derivation of the Erlang-B Formula

The following derivation is based on elementary queueing theory.

The M/M/n/n/∞ queueing model applies to a trunk group of n trunks serving calls arriving with a Poisson distribution. This model is often termed the loss system M/M/n/n/∞ (see figure 5).

Figure 5 - The Loss System $M/M/n/n/\infty$

This model assumes:

- 1) an infinite number of traffic sources,
- 2) Poissonian arrival process with rate λ , i.e.

$$P\{T_A \leq t\} = 1 - e^{-\lambda t}$$
, where T_A is the interarrival time,
- 3) Markovian service process with average service time $\frac{1}{\varepsilon}$, e.g. $P\{T_H \leq t\} = 1 - e^{-\varepsilon t}$, where T_H is the interservice time (note the service time does not have to be exponential; we are simply using the exponential),
- 4) all servers are identical and may be chosen by an arbitrary discipline (homogeneous servers),
- 5) an arriving call is lost if all servers are occupied at the time of arrival,
- 6) lost calls do not influence the behaviour of the system (LCC).

The state diagram for this model may be displayed as a special birth-death process as follows [2]:

States: $x = 0, 1, 2, \dots, n$ (number of occupied servers,
 $x_{\max} = n$),

Transition rates: $\lambda_x = \lambda$, $x = 0, 1, \dots, n-1$
 $\mu_x = x\varepsilon$, $x = 1, 2, \dots, n$

State space and transitions:

The probabilities of state for a birth-death process is

$$P_x = \frac{\lambda_0 \lambda_1 \dots \lambda_{x-1}}{\mu_1 \mu_2 \dots \mu_x} \cdot P_0, \quad 1 \leq x \leq x_{\max} \quad (1)$$

$$P_0 = \left[1 + \sum_{i=1}^{x_{\max}} \frac{\lambda_0 \lambda_1 \dots \lambda_{i-1}}{\mu_1 \mu_2 \dots \mu_i} \right] \quad (2)$$

and inserting $\lambda_x = \lambda$, $\mu_x = x\varepsilon$, in (1) and (2)
 we obtain Erlang's Probability State Distribution for the
 M/M/n/n/∞ Loss System:

$$P_x = \frac{\frac{A^x}{x!}}{\sum_{l=0}^n \frac{A^l}{l!}}, \quad x = 0, 1, \dots, n \quad (3)$$

where $A = \lambda h = \frac{\lambda}{\epsilon}$ is the offered traffic in erlangs.

Equation (3) holds even for loss systems of type M/G/n/n/∞, where G indicates a general distribution of holding times.

Note that the probabilities above may be recursively computed from the balance equations

$$\lambda P_{x-1} + \mu_{x+1} P_{x+1} = (\lambda + \mu_x) P_x$$

which using $\lambda P_0 = \mu_1 P_1$ reduces to

$$\lambda P_{x-1} = \mu_x P_x$$

Now $B = P\{\text{call is lost} \mid \text{call arrives in } \Delta t\}$

$$= \frac{P\{\text{call is lost, call arrives in } \Delta t\}}{P\{\text{call arrives in } \Delta t\}} = \frac{P_n \cdot \lambda \Delta t}{\sum_{i=0}^n P_i \cdot \lambda \Delta t} = P_n$$

and upon application of (3) we obtain Erlang's Loss Formula or what is commonly known as the Erlang-B formula:

$$B = P_n = \frac{\frac{A^n}{n!}}{\sum_{i=0}^n \frac{A^i}{i!}} = B(n, A) .$$

The motivation for choosing the service rate $\mu_x = x\epsilon$ arises from consideration of the distribution of the duration until the first termination of x simultaneous calls being served. The probability of a call lasting longer than t is

$$P\{T_i > t \mid T_i > 0\} = e^{-\epsilon t}, \quad i = 1, 2, \dots, x$$

The duration until the first termination of x simultaneous calls being served is

$$T = \min (T_i)$$

Now,

$$P\{T > t \mid T > 0\} = \prod_{i=1}^x P\{T_i > t \mid T_i > 0\} = e^{-x\epsilon t}$$

and hence by definition (first termination of a call in progress), $\mu_x = x\epsilon$.

3.2 Trunk Group Efficiency

In sizing a trunk group between two points we are concerned with providing an economical number of trunks and still meet service standards. Too few trunks lead to an unacceptable grade of service; too many trunks lead to an excellent grade of service but the capital outlay for the trunks would be unacceptable to the telephone company, which wants to make the most efficient use of its equipment.

The efficiency or occupancy of a trunk group may be defined as the average amount of traffic carried per circuit. For a fixed level of blocking, the efficiency of

the group will increase as the number of trunks increase. This can be seen by using the Erlang-B formula to plot occupancy against number of trunks for a fixed level of blocking (see figure 6A). We referred to this fact in section 2.5, where we described connecting subscribers together as an optimization problem. For a fixed offered load, the efficiency of the group will increase as the number of trunks decrease. This can be seen by using the Erlang-B formula to plot occupancy against number of trunks for a fixed offered load (see figure 6B).

Figure 6A – Occupancy of a Trunk Group at 5% Blocking

Figure 6B — Occupancy of a Trunk Group Offered 120 Erlangs of Traffic

The congestion or blocking of a trunk group with a fixed number of trunks increases with increase in efficiency. This can be seen by using Erlang B to plot blocking against occupancy (or efficiency) as in figure 7.

Figure 7 — Blocking Vs. Occupancy For a Trunk Group of 50 Trunks

3.3 Trunk Efficiency and Alternate Routing

The previous section suggests that one way of increasing the efficiency of a trunk group is to reduce its number of trunks (figure 6B). However, hand-in-hand with this increased efficiency is increased blocking. This means that an increased amount of offered traffic to the trunk group will be lost, i.e. the grade-of-service becomes worse. What we would like is a method for increasing the efficiency of a trunk group while at the same time not allowing the grade-of-service to change for the worse. One method of achieving this is to route traffic through alternate trunk groups. The use of alternate groups for traffic overflowed from congested groups was discussed in chapter 2, section 2.5. We revisit the topic here from the viewpoint of trunk group efficiency. Consider a trunk group between two points A and B as follows:

Suppose we reduce the number of trunks in this group and offer it the same amount of traffic as before. The resulting group would be more efficient (see figure 6B) but more of the offered load would be lost through increased blocking (see figure 7). But suppose, in addition to

reducing the number of trunks in AB, we also introduce an alternate route ATB going through switch T as follows:

Suppose further, that we instruct the switch at A to route traffic blocked on AB via ATB. We have thus increased the efficiency of AB without losing more blocked traffic on AB. There are, however, several points that need clarification:

- i) We have assumed that the alternate groups would be engineered with capacity to carry overflowed traffic from AB.
- ii) The preceding procedure does not aim at abolishing the direct route AB, but rather to make AB more efficient. Without AB, the alternate route would not be "alternate", by definition. Further, the cost per circuit on the alternate route is in general higher than the cost per circuit on the direct or primary route.
- iii) The above mechanism of reducing the number of trunks in the primary group, isolated from the alternate groups, is for conceptual purposes only. In practice, both primary and alternate groups are

engineered together taking into account the circuit cost of primary versus alternate routes. This will be discussed in more detail below.

3.4 The North American Switching Hierarchy

As we saw in chapter 2, section 2.5, there is more than one way to route a call to its destination. Given a moderately large network of say, 50 nodes, the number of ways of routing a call between origin and destination using 48 intermediate nodes is 48! (48 factorial), a very large number indeed. In general, for a total of n nodes, the number of different ways a call can be routed, using k intermediate nodes, is given by

$$\frac{(n-2)!}{(n-2-k)!}$$

This can be seen as follows: the number of distinct combinations of k nodes from $n-2$ intermediate nodes (excluding the 2 end-nodes) is known to be

$${}^{n-2}C_k = \frac{(n-2)!}{(n-2-k)! \cdot k!} .$$

However, for each such combination there are $k!$ permutations or routes. Hence, for n nodes, the number of different ways a call can be routed, using k intermediate nodes, is

$${}^{n-2}C_k \cdot k! = \frac{(n-2)! \cdot k!}{(n-2-k)! \cdot k!} = \frac{(n-2)!}{(n-2-k)!} .$$

Given that there are numerous ways to route calls through a network we need to examine the selection of a routing scheme. Our routing scheme would have to be free of routing cycles (i.e. a call must not be routed to a particular trunk group more than once) and have minimal overhead. The present hierarchical network evolved out of a need to have an orderly, cycle-free routing scheme with minimal overhead.

In order to achieve the routing scheme just mentioned, the switches in a hierarchical network are given different levels or ranks of importance which represent certain restrictions applied to the traffic flow. In the current

North American telephone network, each switch is designated by its position in the switching hierarchy as a class 1,2,3,4, or 5 office (see figure 8). A class n ($n > 1$) switch "homes onto" a class k switch ($k < n$). The group between a switch and its home functions as the first link in the final route for calls originating at the switch. If a call is blocked on this link, it is lost from the network, as there would be no other link to which it could overflow - hence the term "final" route. The class 5 office is termed a local switch since it largely switches local calls. Class 1,2,3, and 4 switches are termed toll switches since they function largely to switch toll calls. Names are given to these switches to reflect their function. For instance, a class 1 office is called a Regional Centre whereas a class 5 office is called an End Office. Names are also assigned to trunk groups depending on the classes of the switches that they connect. Groups connecting class 5 and 4 are termed Toll Connecting trunk groups. Groups between class 1, 2, 3, and 4 offices are called Intertoll trunk groups. There is a further classification of trunks based on how they handle overflow traffic. Groups which do not receive overflow but which may overflow to other trunk groups are termed Primary High Usage (PHU) groups. As the name implies they are usually established for links of high usage or high traffic demand between their end-nodes. Groups which receive overflow and overflow to other groups are called Intermediate High Usage (IHU) groups. Groups which receive overflow but do not overflow to other groups (i.e. any

overflow is lost from the network) are termed Final groups. Note that IHU groups differ from Final groups in that IHU groups pass on their overflow to other groups.

Figure 8 - The North American Switching Hierarchy

The above diagram gives the class numbers, functional names, and standard symbols of switches in the North American telephone switching hierarchy. Sample PHU, IHU, and FINAL trunk groups are also shown. Arrows indicate possible traffic overflow.

3.4.1 Call Routing

Given the above switching hierarchy with associated final groups between switches and their homes, a set of routing rules has been established to take advantage of the potential efficiency of alternate routing. This set of rules is as follows:

Let $H(u)$ be the home of node u ,

$K(u)$ be the class of node u .

The tandem node for node pair (i,j) is:

$$R(i,j) = \begin{cases} 0 & \text{if } H(i)=0 \text{ and } H(j)=0, \\ 0 & \text{if } H(i)=j \text{ or } H(j)=i, \\ H(i) & \text{if } K(i) > K(j), \\ H(j) & \text{otherwise.} \end{cases}$$

Note that $R(i,j)=0$ means that (i,j) is a final group, i.e. no alternate routing is permitted for (i,j) .

For example, consider the 7 node network depicted in figure 9. The routing scheme obtained using the above rules is shown at the bottom of the figure in matrix form. Call this matrix R . Then the entry $R(i,j)=m$ gives the alternate

route (i,m,j) for calls blocked on (i,j) . By reading off entries for each successive direct link in an alternate route, we determine new alternate routes. If we continue this process, and read off entries for the direct links of the new alternate routes in turn, we can see how a call may be routed through the network. We point out that this hierarchy of routes can never lead to a cycle. Calls are routed up the hierarchy to complete on distinct Final groups if neccessary. No call is ever offered to a trunk group which has previously accepted or rejected it.

Figure 9 - Sample 7-Node Routing

7 - Node Network

Routing table for above network:

	1	2	3	4	5	6	7
1	0	0	2	0	4	5	5
2	0	0	0	1	4	5	5
3	2	0	0	2	2	5	5
4	0	1	2	0	0	5	5
5	4	1	2	0	0	0	0
6	5	5	2	5	0	0	5
7	5	5	5	5	0	5	0

Example: Traffic from node 3 to node 5 can be routed through node 2.

3.5 Elements Of Network Sizing

We have been building the framework for the forthcoming section on the mechanics of network sizing. In addition to Erlang-B and alternate routing, there are two more elements that are essential to network sizing. These are 1) the Marginal Load Method, and 2) the Equivalent Random Method (ERM).

3.5.1 The Marginal Load (ML) Method

In the Marginal Load or ML Method for sizing alternate routes we are dealing with economics. Indeed, given the cost of switching and transmission facilities, the telephone company wants to maximize throughput or use of its facilities while maintaining an acceptable grade-of-service. The ML method is one way of approximating this economic goal. The method is applied to a direct route-alternate route triangle and, in conjunction with a blocking formula (such as Erlang-B), determines an economic split of trunks to put on the direct route as opposed to the alternate route. We assume here that the blocking formula used is the Erlang-B formula $B(n, a)$. Let C_1 be the cost per trunk for the direct route and C_2 the cost per trunk for the alternate

route. Assume that C_2 includes the cost of tandem switching. Let " a " be the offered traffic for the period (hour) under consideration. We want to determine the value of n , the number of trunks on the HU group, so that the total cost is minimal. The total cost can be written:

$$COST = C_2 \cdot \frac{a \cdot B(n, a)}{m} + C_1 \cdot n + F$$

where $COST$ is total cost, F is the fixed cost of carrying the background traffic load on the alternate route, and m is the marginal capacity of the alternate route, i.e. the amount of additional traffic that can be offered to the alternate route at fixed blocking, for the addition of one trunk. By "alternate route at fixed blocking" we mean that the blocking of each link in the alternate route remains unchanged. In stating that F is fixed cost we are implicitly assuming that the overflow traffic does not affect the background traffic on the alternate route. In reality, the overflow does affect the background traffic on the alternate route, but the effect is so small that stating F is fixed is a good approximation. The quantity $a \cdot B(n, a)$ is the traffic overflowing to the alternate route and $a \cdot B(n, a) / m$ is the number of additional alternate route trunks needed to carry this overflow. Figure 10A shows the relationship between direct route cost, alternate route cost, and total cost. Treating n as a continuous variable (in practice, n with a decimal fraction may be rounded up or down), the minimum cost is found by setting

$$\frac{\partial \text{COST}}{\partial n} = 0$$

This implies

$$-\frac{\partial (a.B(n,a))}{\partial n} = m \cdot \frac{C1}{C2} = \frac{m}{CR}$$

where $CR=C2/C1$ is termed the cost ratio. The quantity on the left of this last equation is the rate of change of overflow with respect to the number of trunks on the HU group. This is very nearly equal to the "load on the last trunk" defined to be $a(B(n-1,a)-B(n,a))$ if the direct group has n trunks. The quantity on the right is the "economic" load on the last trunk, or economic marginal load (EML). The ML method consists of approximating the minimum cost by sizing the HU group so that the load on the last trunk is equal to its EML, m/CR . The marginal capacity of the alternate route, m , is determined by the desired blocking on the alternate route.

Studies [9] have shown that varying the value used for CR above produces little change in the number of high usage trunks sized. For this reason and to simplify calculations, the following Exponential Approximation [13] for the cost ratio is often used:

$$CR = m / (360 \cdot D^{0.1005})$$

where m is the marginal capacity of the alternate route in erlangs,

D is the straight line distance between the two end nodes in miles.

We need to observe here that in general, the cost per incremental circuit on the alternate route is higher than the cost per incremental circuit on the direct route. This is due to i) extra switching cost at the tandem node, and ii) greater end-node to end-node distance of the alternate route. If we wish to minimize cost, why then would we utilize the alternate route at all? The answer is given by the total cost function above. If we plot the cost of the direct route, the cost of the alternate route, and the total cost, we obtain a plot similar to figure 10B. The point at which the total cost is minimum is given, as we have said above, by

$$\frac{\partial \text{COST}}{\partial n} = 0$$

The fact that the two cost curves cross at the point of minimum total cost allows us to utilize the alternate route.

Figure 10 - Cost Components of the Marginal Load Method

A Principal Costs *B. Route Costs*

3.5.2 Equivalent Random Method (ERM)

Alternate groups receive traffic overflow from the direct groups as well as their own first-offered random traffic. This mixture of random and overflow traffic is not random and a direct application of Erlang-B yields erroneous results (recall that Erlang-B assumes random Poissonian traffic). To determine the number of trunks required on alternate groups we need to calculate the group's blocking, given that its offered traffic is not Poissonian. For the solution of this problem, Bretshneider (1955), and Wilkinson [7] / Riordan (1956), developed the Equivalent Random Method (ERM). The approach taken by these authors was that since Erlang-B required random traffic, why not provide it by constructing a trunk group which would represent the situation at hand, but which would be offered random traffic "equivalent" to the combined overflow and first-offered traffic.

ERM assumes that the combined overflow from a number of HU routes (that receive only Poissonian traffic) is equivalent to the traffic overflowing S full availability trunks to which a random load of A erlangs have been offered (see figure 11). We seek to determine the number of trunks R in the alternate or final route. Y. Rapp [3] gives the following approximate formulae for calculating A and S from the mean and variance of the overflow load:

$$A = v + 3z(z-1)$$

where $z = v/m > 1$ (the "peakedness" of the overflow), v is the variance of the overflow, and m is the mean of the overflow. The number of trunks S is obtained from:

$$S = \frac{A(m+z) - m - 1}{m+z-1}$$

(Rapp obtained these formulae empirically.)

Figure 11 - Components of the Equivalent Random Method

- a = offered traffic to R trunks (peaked traffic)
- a = first-offered traffic + overflow from all groups overflowing to R
- B = Erlang B congestion formula

To determine the number of trunks R , we consider the equivalent group $E=S+R$, to which a random load A is offered, with blocking $B(E,A)$. Note that we can legitimately apply the Erlang-B formula $B(n,a)$ since A is a random Poissonian offered load. Thus by specifying the blocking desired on E , and having calculated A , we can determine E by applying Erlang-B. Then R is simply $E-S$ (S having been determined above).

3.5.2.1 Using ERM in ML

The above discussion shows how one would size the primary high usage groups (PHU's) using the ML method and how one would size the Final groups using ERM. In the latter case we require the blocking on E to be the blocking required for Finals, usually 0.01. What is still unclear is how to size the intermediate high usage groups (IHU's). This is achieved by combining the ML method with ERM. Suppose that we have just sized a PHU using the ML method. Now we would like to size the alternate group which happens to be an IHU group (see figure 12). To do this we consider the IHU group as a direct group with its own associated alternate route. We apply the ML method to this new alternate route triangle, i.e. we want to size the IHU until the load on its last trunk is equal to m/CR . However,

instead of using $a(B(n-1,a)-B(n,a))$ to determine the load on the last trunk, we apply ERM and use instead $A(B(E-1,A)-B(E,A))$. Having determined E in this manner, we find that the number of trunks for the IHU is then $R=E-S$. (Of course, S and A have already been computed using the formulas given above.) It is worthwhile to reiterate here that it would be erroneous to use $a((B(n-1,a)-B(n,a))$ on the IHU because its total offered traffic is not random, its makeup consisting of overflowed non-random traffic as well as its own first offered random traffic.

Figure 12 - Applying ML and ERM to Size IHU Groups

PHU: size using the ML method

FINAL: size using the ER method

IHU: size using ER & ML methods:

1. Obtain S, A from ERM,
2. Set $A[B(E-1, A) - B(E, A)] \neq m/CR$
by adjusting E ,
3. Trunks for IHU = $E - S$.

3.6 Mechanics of Network Sizing

In the following section we give an algorithm used for network sizing that is typically incorporated in a computer program. The algorithm makes use of the preceding notions.

3.6.1 A Sizing Algorithm

As inputs to this sizing algorithm we require:

- . node-to-node (switch-to-switch) traffic demand,
- . switch class, location, and homing information,
- . sizing parameters such as the required blocking on Final groups.

Output will consist principally of the number of trunks required between the switches for the desired grade-of-service as well as meaningful measures of network performance such as the end-to-end blocking. The algorithm consists of the following steps in sequence:

1. Determine the routing using the routing algorithm discussed earlier.
2. Compute the cost ratios (required for the ML Method) This may involve computing the distances between nodes using the switch location information.
3. Use the ML Method to size all PHU groups keeping track of the mean and variances of overflow traffic.
4. Use ERM in ML to size all IHU groups, again keeping track of the means and variances of overflow traffic.
5. Use ERM to size all Final groups to the required blocking level (grade-of-service).
6. Calculate and output the end-to-end blocking.
Output the trunk quantities sized.

In practice, an upward adjustment may be applied to the size of Final groups after step 5, supposedly to account for traffic day-to-day variation. However, this practice has no sound theoretical foundation.

The sizing algorithm just presented is usually implemented with iterative trunk size convergence. A typical iterative scheme is as follows:

1. Estimate the traffic flow;
2. Size the trunk groups to satisfy the design criteria;
3. Calculate the traffic flow;
4. If the design criteria are satisfied, stop;
Otherwise, go to step 2.

Book-keeping overhead to keep track of the means and variances of overflow can be minimized by the use of a suitable node-pair sorting algorithm that would present the node pairs in an optimal sequence for sizing.

3.6.2 Problem Complexity

The above sizing algorithm represents a heuristic solution to the problem of sizing a telephone network. The complexity of the problem does not permit a closed form solution as the number of variables to track grows rapidly with each higher link in the hierarchy. This is evident

when one considers that the offered traffic to each group, has components from every other group which overflows onto it, in addition to the group's own first-offered traffic component. Moreover, the offered traffic to a group will change depending on the sizes of some of the groups that overflow onto it, and the size of a group will be dependent on its offered traffic and so on, ad infinitum.

3.7 Summary

In this chapter we have presented an introduction to teletraffic theory for a Loss System and have indicated how to provision trunks for a telephone voice network. The provisioning presented has been based on the use of one period's traffic demand. In the next chapter we will discuss in detail multi-hour engineering, which uses many periods of traffic to account for time variations in traffic demand.

CHAPTER 4. MULTI-HOUR ENGINEERING

Multi-hour engineering is a way of engineering the network to account for large temporal variations in traffic demand. In this chapter, we are going to explore two existing techniques for multi-hour engineering, namely Sizing-Up (SU) and Extended Sizing-Up (ESU). However, before going into these techniques, we need to discuss two topics: the selection of traffic matrices for multi-hour sizing and the concept of internal versus external point-to-point traffic.

4.1 Selection of Traffic Matrices

In order to represent (or capture) all traffic variation, we must use an astronomically large number of traffic matrices. However, the computational effort required in multi-hour engineering or sizing can become very unwieldy for large numbers of traffic matrices. Fortunately, use of a large number of matrices is not needed since we can effectively represent the traffic variation by "significant" matrices. The problem of selecting these matrices has been

approached by several authors, noteworthy of which is the technique proposed by Beshai, Pound, and Horn [4]. This technique is based on the notion of "proximity" between traffic matrices, where proximity is a measure of the increase in network stress caused by the spatial variation of traffic demand from one period to the next. We do not intend to go into this technique any further. For this work, we simply assume that we have already the traffic matrices that we wish to use.

4.2 Internal and External Point-to-Point Traffic

The size of a particular trunk group is determined by its traffic load, together with the method of design and the design criteria. The traffic load, in turn, is determined by the first offered traffic and the overflow traffic. The overflow traffic is in turn determined largely by the routing scheme.

The traffic load on a trunk group can be termed "internal point-to-point" traffic. For a particular trunk group (i,j) from node i to node j , the internal point-to-point traffic (internal traffic for short) consists of all the traffic flowing along the trunk group (assuming one exists) from node i to node j . This is made up of the

traffic originating at node i destined for node j (the first offered traffic) plus the traffic overflowed onto the group from other groups. This traffic may be switched at node j destined for another node. The external point-to-point traffic of (i,j) (external traffic for short) is the traffic originating at node i destined for node j (the first offered traffic). For PHU groups, the internal and external traffic are the same. Note that the statistical characteristics of the internal and external traffic may differ.

4.3 Sizing Up

The Sizing Up method was proposed by Horn [5] to minimize the computational complexity involved in sizing for a number of traffic periods. In this procedure, the multi-hour sizing problem is broken down into a number of single period sizing problems in which the trunks sized for one period are used as lower bound constraints for the trunks to be sized in the next period. With each additional period, trunk groups may only grow in size or remain unchanged with respect to the trunks sized for the last period; the network is thus sized up. The resultant network is that determined after the last traffic matrix is processed. The SU method gives a first-order solution to the multi-hour sizing problem. The computational effort is

linearly proportional to the number of traffic matrices (assuming equal effort for all matrices). Figure 13 illustrates the SU method.

Figure 13 - Sizing Up

<u>Period</u>	<u>Actual Trunks Sized</u>	<u>Trunks Retained</u>
1	50	50
2	90	90
3	70	90
4	110	110
5	120	120
6	80	120
7	100	120

4.3.1 Rationale of Sizing Up

To understand the process behind SU we use the notions of internal and external traffic. The internal traffic of a network fluctuates according to the variations of external traffic applied through a routing scheme. The SU method captures this fluctuation in internal traffic by sizing each group for its maximum internal traffic. In so doing, it reflects the sharing of trunk resources made possible by alternate routing (recall the definition of the internal traffic). By way of contrast, consider the rather humorous solution of sizing for the maximum external traffic of each node pair, maximized over the number of traffic periods; the result could be an uneconomical oversized network.

4.3.2 Drawback of Sizing Up

The advantages of the SU method are its extreme simplicity and its ease of implementation. The idea of sizing for the maximum internal traffic is sound. However, the method can still result in oversizing. This can come about as follows: The internal traffic at a particular current traffic period reflects the trunks sized in all previous periods. However, this internal traffic fails to

exploit the trunks that will possibly be resized in subsequent periods (assuming the current period is not the last period). This lack of knowledge of the subsequent fluctuation in internal traffic when sizing the current period (other than the last period) results in a gradual build-up in excess trunks. The extent of this build-up depends on the order of processing the traffic matrices.

4.4 Improving the Sizing Up Method

The SU method may be improved to yield higher order solutions at relatively little additional computational effort. Although, the computational effort will be increased, the increase will be at most linear with the number of traffic matrices. We present here two improved modified SU methods. The first, called Extended Sizing Up (ESU) effectively overcomes the drawback of SU described in the previous section. The second, dubbed Sizing Up With Rerouting (SUR), is the proposal of this work. We shall only give an overview of SUR in this chapter, leaving the detailed presentation for chapter 5.

4.4.1 The Extended Sizing Up Method (ESU)

The Extended Sizing Up Method [4] consists of grouping node pairs into traffic decoupled sets and then applying SU to each of these sets in turn. The procedure may result in a significantly more economical network satisfying the design criteria. Before going into ESU in detail, we need to examine the notion of traffic decoupled sets of node pairs.

4.4.1.1 Traffic Decoupled Sets

The idea behind traffic decoupled sets of node pairs is that the traffic flow of one such set does not influence the traffic flow of another such set. We start with some definitions:

1. A node pair (i,j) is said to be independent of a node pair (k,m) if (i,j) 's internal traffic is not affected by the variation of (k,m) 's internal traffic or the size of the trunk group joining k and m .
2. A subset of node pairs is independent of another subset if each member of the first subset is independent of all members of the second subset.

The independence relation just described is not necessarily reciprocal. We suppose that the set S of $n(n-1)$ node pairs belonging to a network with n nodes is divisible into subsets $S(1), S(2), \dots, S(N)$ with the following properties (see figure 14):

- i) The members of each subset are mutually independent.
- ii) Each member in $S(i)$ is independent of the members in its inner subsets $S(i+1), S(i+2), \dots, S(N)$ with $i < N$.
- iii) Each member in $S(i)$ may be strongly dependent on the members in its outer subsets $S(1), S(2), \dots, S(i-1)$, $i > 1$.

We observe that $S(1)$ is totally independent of the remaining subsets, and the innermost subset $S(N)$ may be dependent on all the remaining subsets.

Figure 14 - Traffic Decoupled Subsets of Node Pairs

- ↗ denotes traffic influence (overflow).
- $S(1)$, the outermost subset, is totally independent of $S(2), \dots, S(N)$.
- $S(N)$, the innermost subset, is dependent on $S(1), \dots, S(N-1)$.
- $S(3)$ is independent of $S(4), \dots, S(N)$ but dependent on $S(1)$ and $S(2)$.

4.4.1.2 Extended Sizing Up - Idealized Case

Given the partitioning of node pairs S from the previous section, it can be seen that the internal traffic of a particular $S(i)$ is fully determined (and hence sized) once the internal traffic of its outer subsets (if any) are determined.

We can already write down the procedure for multi-hour sizing making use of S . First we would apply SU to $S(1)$. The internal traffic of $S(1)$ is already fully determined since $S(1)$ has no outer subset on which it is dependent. We then determine the overflow from $S(1)$ for each traffic matrix. This time-varying overflow, together with the time-varying external traffic of $S(2)$ plus the routing scheme then fully determines the internal traffic of $S(2)$. We next apply SU to $S(2)$ and determine the overflow from $S(2)$. This overflow and the overflow from $S(1)$, together with the external traffic of $S(3)$ fully determines the internal traffic of $S(3)$. We apply SU to $S(3)$ and continue in this fashion until we have applied SU to $S(N)$ resulting in a network much leaner in trunks than a straight SU network, and yet satisfying the design criteria.

The computational effort in the procedure just described is comparable to that of SU. Yet, the resulting network is not oversized as in SU. The sizing of each subset $S(i)$ fully exploits the trunks sized for the subsets

on which $S(i)$ depends. However, we have been describing an idealized case: We required each member in $S(i)$ to be totally independent of other members in $S(i)$ as well as being independent of the members in its inner subsets. This ideal situation is mostly not realizable, and in general, we relax the condition somewhat by requiring only that

A member in $S(i)$ may be weakly dependent on other members in $S(i)$, $S(i+1)$, ..., $S(N)$ but may be strongly dependent on members of the outer subsets $S(1)$, $S(2)$, ..., $S(i-1)$.

Under this relaxation, we can give the following algorithm [4] for generating the sets $S(i)$:

1. Initialize an $n \times n$ matrix U such that

$$u(i,j) = \begin{cases} 0 & \text{if } t(i,j)=0 \\ -1 & \text{otherwise} \end{cases}$$

where $t(i,j)$ is the tandem in the routing table T for traffic flow from i to j .

2. Repeat until all $u(i,j)$ (except for $i=j$) are positive:

For $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$, let $m=t(i,j)$,

If $u(i,m) \geq 0$ and $u(m,j) \geq 0$, then

$$u(i,j) \leftarrow u(i,m) + u(m,j) + 1$$

The resultant number of subsets is $N = \max_{i,j} u(i,j) + 1$

with (i,j) being a member of $S(k)$, $k = N + 1 - u(i,j)$. Note that $t(i,j)=0$ indicates no tandem available for (i,j) , i.e. (i,j) is a Final group. Further, the number of subsets N depends on T , the routing table.

4.4.1.3 Extended Sizing Up -Algorithm

The necessary relaxation discussed in the previous section leads to increased computational effort in the multi-hour sizing. No longer can a set $S(i)$ be sized once its outer subsets have been sized. It is now necessary to take account of the influence (weak but there all the same) of $S(i)$'s inner subsets. The following Extended Sizing Up algorithm has been proposed by Beshai et al [4]:

Let $C(r,i)$ denote the capacities (trunk group sizes) of subset $S(i)$, $1 \leq i \leq N$, after completion of step r in the following iterative procedure: Let $C(0,i)$ denote the initial (pre-sizing) capacities.

1. Using $C(0,1)$, $C(0,2)$, ..., $C(0,N)$ as lower bound, apply SU to the entire network. After doing this, reset the capacities of $S(2)$ to $S(N)$ to their initial values $C(0,2)$ to $C(0,N)$. Retain only the capacity of $S(1)$, i.e. $C(1,1)$.
2. Using $C(1,1)$, $C(0,2)$, ..., $C(0,N)$ as lower bound, apply SU to the entire network again. After doing this, reset the capacities of $S(3)$ to $S(N)$ to the initial values $C(0,3)$ to $C(0,N)$. Retain only the capacities of $S(1)$ and $S(2)$, i.e. $C(2,1)$ and $C(2,2)$.

Proceeding in this manner, and upon completion of step $r < N$, the capacities of subsets $S(r+1)$ to $S(N)$ are reset to their initial values and the capacities $C(r,1)$, $C(r,2)$, ..., $C(r,r)$, $C(0,r+1)$, ..., $C(0,N)$ are used as lower bounds in step $r+1$.

Figure 15 illustrates this procedure. We note that unlike the idealized case, the overflow need not be saved between steps of sizing for $S(i)$. In this procedure, the entire network is re-sized in each step. The internal traffic is thus recalculated with each step.

Figure 15 - Extended Sizing Up

Each concentric layer is a traffic decoupled subset of node pairs. The above diagrams represent a network with 3 such layers.

A. Initial pre-sizing capacities.

B. All layers sized up for all time periods; sized-up capacities of layer 1 retained; layers 2 and 3 reset to initial capacities.

C. All layers sized-up for all time periods; sized-up capacities of layer 2 retained, together with layer 1; layer 3 reset to initial capacities.

D. All layers sized-up for all time periods; sized-up capacities of layer 3 retained, together with layers 1 and 2. ESU complete.

OF / DE

The Extended Sizing Up method has been implemented in a highly efficient software package. More details on this implementation may be obtained by consulting Beshai et al [4].

4.4.2 Sizing Up With Rerouting (SUR)

We intend here to give just an overview of Sizing Up With Rerouting, as Chapter 5 describes it in detail. The basic idea behind this method is very simple: When introducing a new traffic period in the SU procedure, limit the establishment of new trunks by rerouting traffic to utilize idle trunks already sized for a previous period. The idle trunks may arise due to the time varying external traffic, i.e. trunks sized for (i,j) in period k may be idle in period m (assuming period k sized before period m) because the internal traffic for (i,j) has decreased in period m . The SUR procedure is first to derive a set of new routing tables for rerouting traffic to existing idle groups (one for each new traffic period) and then to replace the fixed hierarchical routing with the new routing for each period to be sized in the SU procedure. The SUR procedure improves over SU in producing a more economical network satisfying the design criteria with fewer trunks. The new routing

tables derived in SUR are to be implemented in switches capable of routing control by change in routing table (most modern switches have this capability). The main advantages of SUR are its simplicity and minimal extra computational effort when compared to SU.

4.5 Summary

In this chapter we have dealt with the mechanics of multi-hour engineering, specifically with the techniques of Sizing Up and Extended Sizing Up, together with their advantages and disadvantages. We concluded with a brief overview of Sizing Up With Rerouting, which we propose as an improvement over Sizing Up. In chapter 5 we will present the SUR procedure in detail.

CHAPTER 5. SIZING UP WITH REROUTING (SUR)

Sizing Up With Rerouting was mentioned briefly in the previous chapter as an improvement over Sizing Up that yields a more economical network by way of fewer trunks required to satisfy the design criteria. In this chapter we will present SUR in detail. We will begin by relating the temporal variations in multihour traffic with trunk occupancy. We will then look at how rerouting can increase the utilization of low occupancy groups in a new period and thereby ward-off the establishment of new trunks. Next, we will present and analyze the rerouting algorithm at the heart of SUR. We will complete the chapter by giving a step-by-step presentation of the SUR method.

5.1 Temporal Traffic Variations and Trunk Occupancy

The method of Sizing Up sizes trunk groups according to their internal traffic levels. However, due to the temporal variations of multihour external traffic, the internal traffic level on a particular trunk group can change from one period to the next period. Thus, even though the

internal traffic of (i,j) may warrant the establishment of a trunk group from i to j in a particular period, that same group may have very low traffic occupancy in the next period. Conversely, the internal traffic for (k,m) may be insignificant for a particular period but then increase sufficiently in the next period to justify the establishment of new trunks from k to m in that period. This leads to the following situation with each new period: there are groups which are idle or have very low traffic occupancy due to decreases in the internal traffic for those groups, and there are new trunks established corresponding to those node pairs experiencing sufficient increases in internal traffic. Figure 16 depicts the situation just described. A more economical network could be sized if we could reduce the establishment of new trunks by making use of the spare capacity of already established groups (i.e. groups with low traffic occupancy). This is the basic idea behind SUR.

Figure 16 - Effects of Internal Traffic Variations in SU

Typical variation of internal traffic and required trunks for a single trunk group:

Legend:

- internal traffic
- required trunks based on the internal traffic
- difference between required trunks and sized up trunks - excess capacity which may be exploited by rerouting

5.2 Utilization of Idle Trunk Capacity Through Rerouting

The mechanism through which we can make use of idle or low occupancy groups in a new period, and thence to reduce the number of new trunks established is rerouting. Basically, we reroute traffic away from potential new trunks to existing idle trunks. To reroute, we look at alternate route triangles where the base of the triangle is the direct route and the other two sides comprise the alternate route. Specifically, if we have nodes i , t , and j , where (i,j) is the direct route and (i,t,j) the alternate route, to reroute means we seek a new tandem node T where (i,T,j) is the new alternate route for (i,j) , as follows:

5.2.1 Criteria for Rerouting

At this point it is beneficial to examine i) how we could choose a candidate routing triangle for rerouting, and ii) how we could find a new tandem node T for the new alternate route. The objective in SUR for rerouting traffic is to reduce the number of new trunks sized in a new period. In an existing routing triangle (i, t, j) new trunks are sized for the links defined by (i, t, j) due to increases in the internal traffic levels of the links. Contributions to these increases in internal traffic are due to the associated overflow and point-to-point (first offered) traffic loads. For a particular routing triangle we are interested in increases in the offered traffic of the direct route in the new period (the alternate links will be considered for traffic increase when they are the direct routes in their own respective routing triangles). This increase in traffic offered to the direct route will lead to increases in the internal traffic loads of both the direct route and the alternate route and result in new trunks established for either the direct route, the alternate route, or both (measurements have shown that very few calls use more than one routing triangle; it is thus adequate to consider one routing triangle per direct route). To find a new tandem node T we require (i, T, j) to have greater spare capacity than (i, t, j) in the new period, i.e. the internal traffic levels of (i, T) and (T, j) must have decreased more

than the internal traffic levels of (i,t) and (t,j) . Note that $T=t$ is possible (no rerouting). The criteria for rerouting are then:

In the new period, reroute if

1. the direct route (i,j) of the routing triangle (i,t,j) has an increase in offered traffic, and
2. there is some other alternate route (i,T,j) with a larger decrease in internal traffic than (i,t,j) .

5.2.2 Approximate Criteria for Rerouting

The above criteria for rerouting requires examinations of offered and internal traffic levels. To implement this strategy in an algorithm for rerouting is very complex, due to the high correlation of the variables involved. As soon as we effect one reroute, the offered and internal traffic levels of groups that receive overflow from the present direct route (whose alternate route we have just changed) are all affected, necessitating a recomputation of the relevant traffic loads of those groups. To do this for every reroute would entail an unacceptably large amount of

computational effort. We adopt instead the following heuristic: we examine trunks sized rather than offered and internal traffic levels. The new criteria for rerouting are then:

In the new period, with the network sized assuming no routing changes, reroute if

1. the direct route (i,j) of the routing triangle (i,t,j) has an increase in trunks, and
2. there is some other alternate route (i,T,j) with a larger decrease in trunks than (i,t,j) .

The efficacy of this heuristic stems from the fact that a trunk group is sized for its internal traffic level. The size of the trunk group is therefore a good approximation of the internal traffic between its two endpoints. The second condition allows rerouting to take place even if in the new period the original alternate route shows a decrease in number of trunks - as long as some other alternate route has an even greater decrease in trunks. This condition increases the likelihood that the new alternate route has enough spare capacity to handle the traffic increase of the direct route. Note that the heuristic works even for those direct routes which have no increase in first-offered traffic but do show increases in overflowed traffic in the new period (since the

size of a trunk group reflects also the amount of overflowed traffic). Implied in the second condition above is the method for seeking the new tandem node T : we seek T such that (i,T) and (T,j) show the largest common decrease in trunks in the new period from among all possible candidates T .

The above approximation allows the derivation of a straightforward rerouting algorithm based on the comparison of trunk matrices alone, without the need to explicitly flow the traffic through the network and recompute traffic levels with each reroute effected. However, the internal traffic is regenerated with each resizing step in the SUR algorithm below.

5.3 The Rerouting Algorithm

We begin with some notation:

Let

- T1 - trunk matrix corresponding to the traffic matrix of highest traffic volume over all periods;
- T - trunk matrix for the same period as the new routing table we wish to derive;
- R - fixed hierarchical routing table derived from the network file as explained in chapter 3;
- S - matrix of traffic decoupled sets as explained in section 4.4.1.1 and generated by the algorithm given in section 4.4.1.2 ; the node pair (i,j) is in set $s(i,j)$, where $s(i,j)$ is the ij th element of S;
- R' - new routing table generated by the Rerouting Algorithm;
- $a(i,j)$ - the ij th element of the matrix A;
- n - the number of nodes in the network;
- k - the current set of node pairs in S currently being processed;
- m - the number of traffic decoupled sets in S;

μ - "tandem type" to be explained below, possible values are 1,2, and 3;

LIST - list of node pairs (i,j) that show increases in trunks for the current period;

CHLOOP - a procedure for checking if a proposed routing change will result in a routing loop; the method used for this check is described in the next section.

The Algorithm has the following inputs and outputs:

Inputs: T_1 , T , R , S , k , m , n

Output: R'

A flowchart of the algorithm is presented as figure 17.

The Algorithm itself follows (Pidgin ALGOL):

1. Set $r'(i,j) \leftarrow r(i,j)$ for all i,j .
 If $k = m$, go to step 5. Otherwise,
 compute $D \leftarrow T - T_1$ for all i,j .

2. Do the following:

```

begin;
  IDX <-- 0;
  for i <-- 1 step 1 until n do;
    for j <-- 1 step 1 until n do;
      if s(i,j) = k then do;
        if d(i,j) > 0 then do;
          IDX <-- IDX + 1;
          LIST(IDX) <-- (i,j);
          TINCR(IDX) <-- d(i,j)/2;
        end;
      end;
    end;
  end;
  LISTSIZE <-- IDX;
end;

```

3. Sort LIST in descending order of TINCR giving LIST' and TINCR'. Then LIST'(1) gives the node pair with the largest increase in trunks (TINCR'(1)) for the current period. LIST'(LISTSIZE) gives the node pair with the smallest increase in trunks for the current period (TINCR'(LISTSIZE)).

4. Do the following:

```

begin;
  for  $\mu$  <-- 1 step 1 until 3 do;
    for IDX <-- 1 step 1 until LISTSIZE do;
      (i,j) <-- LIST'(IDX);
      BIGD <-- TINCR'(IDX);
      if BIGD > 0 then do;
        MAXT <-- -99999;
        DELT <-- -99999;
        TANDMAX <-- 0;
        for TAND <-- 1 step 1 until n do;
          if (s(i,TAND) > k) and (s(TAND,j) > k) then do;
            if ( $\mu$ =1) and
              (d(i,TAND) < 0 and d(TAND,j) < 0) then do;
              DELT <-- MIN (-d(i,TAND), -d(TAND,j));
            end;
            if ( $\mu$ =2) and
              (d(i,TAND) < 0 or d(TAND,j) < 0) then do;
              DELT <-- MIN (d(i,TAND), d(TAND,j));
            end;
            if ( $\mu$ =3) and
              (d(i,TAND)  $\geq$  0 and d(TAND,j)  $\geq$  0) then do;
              DELT <-- MIN (d(i,TAND), d(TAND,j));
            end;
            if DELT > MAXT then do;
              RLOOP <-- CHLOOP;
              if RLOOP is 'false' then do;
                MAXT <-- DELT;
                TANDMAX <-- TAND;
              end;
            end;
          end;
        end;
      end;
      if (TANDMAX > 0) then do;
        r'(i,j) <-- TANDMAX;
        TINCR'(IDX) <-- -BIGD;
        d(i,TANDMAX) <-- d(i,TANDMAX) + BIGD;
        d(TANDMAX,j) <-- d(TANDMAX,j) + BIGD;
        d(j,TANDMAX) <-- d(j,TANDMAX) + BIGD;
        d(TANDMAX,i) <-- d(TANDMAX,i) + BIGD;
      end;
    end;
  end;
end;
end;
end;
end;

```

5. Output R' and stop.

Figure 17 - Flowchart of the Rerouting Algorithm

5.3.1 Routing Loops

A routing loop is a routing anomaly in which traffic is routed back to its origin, i.e. a tandem node becomes a point of origin or destination. This, of course, cannot be allowed to occur, as it serves no useful purpose and is wasteful of resources. Routing loops are automatically ruled out when the routing table is derived with the algorithm given in section 3.4.1. However, routing loops can inadvertently be introduced in user provided routing tables or in the Rerouting Algorithm above, if routing loop checks are not performed. The following routing loop check is used in the Rerouting Algorithm and is based on the fact that traffic flowing in a loop can never encounter a FINAL group:

```
Routing Loop Check for  $r'(i,j)$   $\leftarrow$  TANDMAX
      (in Pidgin ALGOL)
```

```
Procedure CHLOOP( $R'$ , TANDMAX,  $i, j$ );
begin;
  CHLOOP  $\leftarrow$  'false';
  TDM  $\leftarrow$  TANDMAX;
   $r'(i,j)$   $\leftarrow$  TANDMAX;
  for CTR  $\leftarrow$  1 step 1 until 8 do;
    TDM  $\leftarrow$   $r'(i, TDM)$ ;
    If TDM < 1 then return;
  end;
  CHLOOP  $\leftarrow$  'true';
  return;
end;
```

The value 8 is a nominal value based on the network topology. Since it is rare to encounter a telephone switching hierarchy with more than, say 4 tandeming levels before encountering a FINAL group, the value 8 for the above

check is more than adequate. The "if" condition is for the detection of a FINAL ($r'(i,j)=0$ for a FINAL).

5.3.2 Analysis of the Rerouting Algorithm

The Algorithm compares the trunks sized for a current period against the trunks sized for the "first" period. The current period is the interval for which we wish to derive a new routing table. The "first" period has (or can be assigned) the heaviest total traffic of all the periods. This increases the likelihood of spare trunk capacity in subsequent periods, since the matrix with the heaviest total traffic volume is sized first. Direct routes with increases in trunks for the current period are evidenced by $d(i,j) > 0$. Tandem nodes on alternate routes with spare capacity are evidenced by:

1. $d(i,TAND) < 0$ and $d(TAND,j) < 0$ ($\mu=1$)
2. $d(i,TAND) < 0$ or $d(TAND,j) < 0$ ($\mu=2$)

Tandem nodes (TAND) on alternate routes with no spare capacity are evidenced by:

3. $d(i,TAND) \geq 0$ and $d(TAND,j) \geq 0$ ($\mu=3$)

The Rerouting Algorithm processes the above tandem types in order: first, type $\mu=1$, then type $\mu=2$, and finally type $\mu=3$. Only if it is not possible to find a particular tandem type for a direct route is the next tandem type sought after. For type $\mu=1$, both alternate route links have spare capacity. For type $\mu=2$, only one of the alternate route links has spare capacity. The routing of traffic through a type $\mu=2$ node will make use of the link with spare capacity and grow the link that does not have spare capacity. However, the grown link may be rerouted upon processing of the next set in S (set $k+1$) which receives overflow from set k . Category $\mu=3$ is included basically for two reasons: i) to propagate the trunk increase of the direct route to the next set in S (set $k+1$), as in the case of the link with no spare capacity for type $\mu=2$, and ii) the larger trunk groups resulting from such a reroute would be more efficient and hence fewer trunks would be needed overall. Reason i) increases the probability that a reroute making use of spare capacity would be found for the direct route in the current set, since the next tandeming set ($k+2$) may have surplus spare capacity. This reasoning applies also to type $\mu=2$ for the link without spare capacity. Figure 18 illustrates the above three tandem types.

Figure 18 - The Three Tandem Types in the Rerouting Algorithm

The 3 Tandem Types:

$\mu = 1$: $D(i, T) < 0$ and $D(T, j) < 0$

$\mu = 2$: $D(i, T) < 0$ or $D(T, j) < 0$

$\mu = 3$: $D(i, T) \geq 0$ and $D(T, j) \geq 0$

For type $\mu=1$ the common spare capacity of the alternate route is found by taking the minimum of $-d(i,TAND)$ and $-d(TAND,j)$. For types $\mu=2$ and 3 , the common "spare" capacity of the alternate route is defined to be the minimum of $d(i,TAND)$ and $d(TAND,j)$. For type $\mu=2$, this definition yields the spare capacity of the link with spare capacity. For type $\mu=3$, the definition obtains the alternate route with the largest number of trunks for the link with the smaller trunk increase. The total trunk increase of the direct route to be rerouted is applied to the "spare" capacity of the alternate route, regardless of whether there is sufficient spare capacity to accomodate the increase. In the case of insufficient alternate route spare capacity, the excess trunks transferred will be propagated to the next traffic decoupled set as explained in the description of type $\mu=3$ above.

Each time a reroute is made the matrix D must be updated with the amount of transferred trunks in order to keep track of remaining spare capacity. Note that it is not necessary to update $d(i,j)$ since $d(i,j)$ is never used a second time in the Algorithm.

Sorting the node pairs (i,j) results in a better match of node pairs with larger trunk increases to alternate routes with larger spare capacities (a more optimal

allocation of spare capacities). As a possible alternative, one might consider allocating node pairs to alternate route spare capacities on a one-to-many basis, but this is not possible here, since the relationship between node pairs and alternate routes has to be one-to-one.

We now analyze the Algorithm for its time complexity. The time complexity $T(n)$ is said to be $O(f(n))$ (read order $f(n)$) if $T(n) \leq cf(n)$ for some constant c and all but some finite (possibly empty) set of nonnegative values of n . It is easy to see from this definition that if $T(n)$ is $O(f(n))$ and $S(n)$ is $O(f(n))$, then $T(n) + S(n)$ is also $O(f(n))$. It is also evident that if $T(n)$ is $O(f(n))$, $S(n)$ is $O(g(n))$, and $g(n) \leq f(n)$, then $T(n) + S(n)$ is $O(f(n))$. Also, if $T(n)$ is $O(f(n))$ and $f(n)$ is $O(g(n))$ then $T(n)$ is $O(g(n))$.

Let n be the number of nodes in the network,

p be the number of final links,

m be the number of node pairs with increased trunks found in step 2.

Then we have the following worst case time complexity per step:

Step	Operations	Time Complexity
----	-----	-----
1	n^2 subtractions, 1 comparison, $2n^2$ assignments	$O(n^2)$
2	n^2 additions, n^2 divisions, $2n^2$ comparisons, $3n^2 + 2$ assignments	$O(n^2)$
3	Bubble Sort of m elements	$O(n^2)$
4	$3m(10n+1)$ comparisons $3m(2n+1)$ subtractions $12m$ additions $3m(4n+10)$ assignments	$O(n^3)$
5	n^2 print operations	$O(n^2)$

The time complexities of steps 3 and 4 are arrived at as follows:

In the worst case $m = n(n-1) - p < n(n-1) < n \cdot n = n^2$.

Since a bubble sort of m elements is $O(m^2)$, it can

be seen that step 3 is $O(n^4)$. The time complexity of step 4 can be seen by simply replacing m with n^2 since $m < n^2$, as shown above.

The time complexity (worst case) of the Rerouting Algorithm is therefore $O(n^4)$ (obtained by adding the time complexities of steps 1 to 5). Note that by dispensing with the bubble sort the time complexity reduces to $O(n^3)$.

5.4 Sizing Up With Rerouting: Algorithm

Finally, we present the SUR Algorithm (see figure 19 for a flowchart):

Suppose we have n traffic matrices with one matrix for each period to be used in multihour engineering.

Let A_i be the traffic matrix for period i ;

$N(A_i)$ be the sum of all the elements of A_i ;

R be the fixed hierarchical routing table;

R_{ik} be the routing table generated by the

Rerouting Algorithm after processing
for traffic decoupled set k in period i ;

S_{ik} be the matrix of traffic decoupled sets
generated after rerouting set k in
period i ;

$s_{ik}(i,j)$ be the ij th element of the matrix S_{ik} ;

S be the matrix of traffic decoupled sets
generated using R ;

NUMSETS be the number of traffic decoupled sets in S ;

T_{ik} be the trunks matrix after sizing for
period i using A_i and R_{ik} ;

LISTSU be a list of 2-tuples (A_i, R_{im}) , $i=1,2,\dots,n$,
 $m=\text{NUMSETS}-1$, for use in the sizing-up
algorithm;

Further, let $\text{REROUTE}(\text{input/output})$ denote the Rerouting
Algorithm, $\text{SIZE}(\text{input/output})$ denote the sizing algorithm,
 $\text{SU}(\text{input/output})$ denote the sizing-up algorithm, and
 $\text{DECOUP}(\text{input/output})$ denote the algorithm for generating
 S_{ik} .

We assume $N(A_1) \geq N(A_2) \geq \dots \geq N(A_n)$ (if not, we can order and reassign the A_i to periods so that this ordering holds). The SUR Algorithm follows:

SUR Algorithm

```

begin;
  /* Initialization */
  m <-- NUMSETS-1;
  LISTSU <-- (A1,R);
  for i=1 step 1 until n do;
    Ri0 <-- R;
    Si0 <-- S;
    Ti0 <-- SIZE(Ai,R);
  end;
  for i=2 step 1 until n do;
    p <-- i-1;
    /* Derive new routing table for period i */
    for j=1 step 1 until m do;
      k <-- j-1;
      Rij <-- REROUTE(Tp0,Tik,Rik,Sik,j);
      Sij <-- DECOUP(Rij);
      m <-- sij(1,1) - 1;
      if j < m then do;
        Tij <-- SIZE(Ai,Rij);
      end;
    end;
    /* Size-up using new routing tables */
    LISTSU <-- LISTSU + (Ai,Rim);
    Ti0 <-- SU(LISTSU);
  end;
end;

```

The trunks for the SUR sized network is given by $Tn0$.

Figure 19 - Flowchart of the SUR Algorithm

5.4.1 Explanation of the SUR Algorithm

At the start, the method does some initialization of variables and sizes the network for each of the n periods of traffic, yielding n trunk matrices. The sizing done here is the usual single period sizing. The resultant trunk matrices will be used in the Rerouting Algorithm.

After initialization, the Rerouting Algorithm is alternatively applied to each traffic decoupled set starting with the first set until the second to last set. It is not possible to apply the Algorithm to the last set, since node pairs in that set are connected by FINALS (which have no alternate routes, by definition). The new routing table is used in the sizing algorithm to re-size the network to reflect the last change in routing. The whole idea here is to apply the Rerouting Algorithm effectively. This means that the trunks compared in the Algorithm must closely reflect the internal traffic. To achieve this, the trunks for the next traffic decoupled set for application in the Rerouting Algorithm, are re-sized using the routing table generated for the current set. The internal traffic of the next traffic decoupled set consists of the first offered traffic and any overflow from the current set. The distribution of overflow to the next set depends on the routing of the current set. Since we have changed the

routing of the current set, it is necessary to re-size the next set using the changed routing.

The Beshai algorithm (DECOUP) is applied to the new routing table produced by REROUTE. This is necessary since the matrix of traffic decoupled sets depends on the make-up of the routing table. Changes in the routing table may mean changes in the matrix of traffic decoupled sets.

Note that the Rerouting Algorithm, as used here, compares the trunks of a new period $i > 2$ to the trunks sized-up for all the previous periods (sized-up using the available new routing tables from REROUTE).

5.5 Summary

We have in this chapter related temporal traffic variations and trunk occupancy and from there explained how trunks with low occupancy may be re-utilized to reduce the number of new trunks sized in multihour engineering. The mechanism for this reuse is the rerouting of traffic to low occupancy trunks. We then examined the Rerouting Algorithm which incorporates the mechanism just mentioned. We closed the chapter by describing the SUR Algorithm which

incorporates and uses the Rerouting Algorithm. The SUR algorithm represents an improvement over SU.

In chapter 6 we will give the results of using SUR for the multihour sizing of two networks and compare these results to those obtained by the use of SU.

CHAPTER 6. SIZING UP WITH REROUTING: EXPERIMENTAL RESULTS

Sizing Up and Sizing Up with Rerouting were applied to two networks representative of existing networks. The results of SU and SUR were then tabulated and compared. In this chapter, we will present these results. We first give some details of the two test networks.

6.1 Test Networks

The two test networks were labeled Network 34 and Network 152.

Network 34 is a representative network containing 34 nodes and spanning one time zone. Three hours of observed traffic data for this network were available and used. Figure 20 gives the nodes distribution for Network 34. The traffic and routing data for Network 34 have not been included in this document due to their large volume.

FIGURE 20 - NETWORK 34 NODES
DISTRIBUTION

The traffic matrices for network 34 covered the following time periods:

<u>Traffic Matrix</u>	<u>Time</u>
341	10-11 AM
342	2- 3 PM
343	6- 7 PM

Network 152 is another representative network consisting of 152 nodes and spanning 5 time zones. There is a one hour difference between each time zone. This means that the network has a different busy hour for each time zone, which partially contributes to the differences in trunk occupancy throughout the entire network. These heightened differences in trunk utilization are just what SUR can exploit to reduce the number of new trunks sized. Figure 21 gives the nodes distribution for Network 152.

FIGURE 21 - NETWORK 52 NODES
DISTRIBUTION

All four traffic matrices available for this network were used. Again, the traffic and routing data for this network have not been included in this document due to their large volume. The four traffic matrices cover the following hours (Central Standard Time):

<u>Traffic Matrix</u>	<u>Time</u>
1521	9-10 AM
1522	1- 2 PM
1523	5- 6 PM
1524	8- 9 PM

The above networks were used primarily due to the fact that data for them was readily available. The writer would have liked to apply SUR to a large private network spanning many time zones, where the greater traffic volatility should result in greater trunk savings through SUR. Unfortunately data for such a network was not available.

6.2 Computer Runs

The computer runs were done on an IBM 3084 running VM/SP. The sizing program used applied the network sizing methodology described in chapter 3. In addition, the program had the capability to do SU using the same routing

table for each hour. For SUR, it was only necessary to modify it to accept a new routing table for each new hour. We shall not go into the operational details of running the program. In each single hour sizing problem, the two networks were sized using the methodology described in chapter 3, to design criteria of 1% blocking on FINAL groups, a marginal load of 30 CCS or 0.83 erlangs, and using the Exponential Approximation for cost ratios.

6.3 Experimental Results

We now give the experimental results obtained for the two test networks in terms of cost and trunk savings achieved.

The cost figures shown in the results sections below were provided by the sizing program. They are based on the following approximate relative cost units:

cost per trunk-mile (short distance, < 501 miles)	1.5
cost per trunk-mile (long distance, > 500 miles)	1.0
cost per trunk - terminations	52.0
cost of switching (per trunk)	36.0

The cost per trunk-mile is the per mile cost of a physical trunk. The cost per trunk - terminations, is the cost of terminating a trunk on a switch multiplied by 2 for the 2 trunk ends.

6.3.1 Network 34 Results

The following results were obtained for Network 34:

Traffic Matrix	Traffic Volume (Erl)	Trunks Sized SU	Trunks Sized SUR	Trunks Saved SUR
341	8673.75	10809	10809	-
342	7510.72	10891	10884	7
343	7114.14	11368	11261	107
Cost of SU network:		2,880,500		
Cost of SUR network:		2,848,000		
Cost Saving by SUR :		32,500		

6.3.2 Network 152 Results

The following results were obtained for network 152:

Traffic Matrix	Traffic Volume (Erl)	Trunks Sized SU	Trunks Sized SUR	Trunks Saved SUR
1521	69228.87	86757	86757	-
1522	59506.25	88194	88054	140
1523	52082.61	92690	92166	524
1524	51355.53	94306	93819	487
Cost of SU network:		38,175,000		
Cost of SUR network:		37,625,000		
Cost Saving by SUR :		550,000		

It is seen that with the processing of the 1524 traffic matrix the trunks saved by SUR actually decreases. This appears paradoxical at first but it can be easily explained as follows. Up to and including the 1523 traffic there has always been sufficient spare capacity to realize a trunk saving with SUR. It appears that after processing the 1523 traffic, spare trunk capacity in the network was almost completely used up. Consequently, with the processing of the 1524 traffic, it was necessary to "break new ground" as it were, and establish new trunks. This explanation is even more plausible when one considers that the 1524 traffic

represents the evening residential peak in calls, which may have an entirely different pattern from the morning business peak.

6.4 Distribution of Trunks: SU vs. SUR

It is interesting to compare the distribution of trunks among PHU, IHU, and FINAL groups between SU and SUR. We have the following results for the sized-up Network 34:

Type	Trunk Groups		Number of Trunks		Av.Trunks/Group	
	SU	SUR	SU	SUR	SU	SUR
PHU	162	162	1736	1736	10.72	10.72
IHU	192	192	5029	4973	26.19	25.90
FINAL	33	33	4603	4552	139.48	137.94

The following results were obtained for the sized-up network 152:

Type	Trunk Groups		Number of Trunks		Av.Trunks/Group	
	SU	SUR	SU	SUR	SU	SUR
PHU	1849	1216	22342	12736	12.08	10.47
IHU	2139	2760	43036	52234	20.11	18.92
FINAL	151	151	28928	28849	191.58	191.05

We see that in both test networks, SUR has reduced the average size of groups, although not by too much in the case of FINALS for Network 152. We also observe that for Network 152 SUR has shifted trunks from PHUs to IHUs (both groups and trunks decreased for PHUs and increased for IHUs). The majority of trunk savings for Network 152 has apparently been achieved by the rerouting of traffic from PHUs to spare capacity among IHUs. This was not the case for Network 34, where PHU trunks were the same for both SU and SUR. There, the savings occurred through an almost even reduction in trunks for IHU and FINAL groups. This difference in the way trunk savings were achieved is due principally to the fact that Network 152 spans 5 time zones whereas Network 34 spans a single time zone. The wider geographical span and range of busy hours for Network 152 allowed more reroutes to occur at the PHU level. In other words, spare capacity in the case of Network 152 occurred in a more horizontal fashion (across the switch homing hierarchy) rather than in a vertical fashion (up the switch homing hierarchy). Apparently, the manner in which SUR savings are achieved is influenced by the network architecture.

6.5 Summary

We have presented and compared the results of SU and SUR applied to two representative test networks. It was found that savings in trunks and capital cost could be achieved through the use of SUR in multi-hour engineering. In the next chapter we will present our conclusions and suggestions for future work based on the present experience with SUR.

CHAPTER 7. SIZING UP WITH REROUTING: CONCLUSIONS

We began this thesis by introducing the reader to the traffic theory that stands behind the common telephone. We proceeded to explain how a telephone network is sized for the number of required trunks, in order to provide an acceptable grade-of-service. Then we introduced the idea of multihour engineering to account for temporal variations in the external traffic. We compared various strategies for multihour engineering and then proposed our own method, namely Sizing Up with Rerouting.

The main objective of this work was to demonstrate that a method can be constructed for multihour engineering that would improve on the Sizing Up method by making use of idle capacity. To the extent that we have constructed SUR and used it on two representative test networks, we have achieved this goal.

In this chapter, we present some conclusions based on our experience with SUR. We will also present some suggestions for future work.

7.1 Conclusions on SUR

Based on the test runs we have done using SUR, we conclude that SUR is a viable method of multihour engineering a network. The idea of rerouting to make use of idle trunk capacity in order to reduce the total number of trunks required is sound. To support this notion, SUR achieved a trunk saving of approximately 1% over SU. This trunk saving will vary depending on network topology and the pattern and intensity of traffic flow. The cost savings allowed by SUR are more impressive. As we saw for the two test networks the cost savings of SUR when compared to SU were approximately 32,000 and 550,000 in terms of relative cost units (transmission cost per trunk mile = 1, for distances greater than 500 miles). These savings can be had without any extra capital outlay, by merely implementing the appropriate routing table generated by SUR for the corresponding time period. Of course, when we make this statement, we assume that the switch can easily accept a new routing table. At the present time all digital switches have this capability and before long all switches will be digital. Certainly, new networks will use digital switching. The applicability of SUR is not a problem.

The SUR algorithm does not require any new data other than the usual data required for sizing a network. Existing single-period sizing programs or SU programs can easily be

adapted for SUR. The author was able to do this with minimal effort. The only new elements required are the Rerouting Algorithm and the Beshai Algorithm for generating traffic decoupled sets. Neither of the latter two algorithms are difficult to code (FORTRAN implementations of the Rerouting Algorithm and the Beshai Algorithm are available from the author).

7.2 Future Work

We offer the following suggestions for future work:

The Rerouting Algorithm .

The Rerouting Algorithm as presented in chapter 5 has not been optimized for efficiency (optimization was not the primary goal). By judicious experimentation and innovation the time complexity of the algorithm may definitely be improved. For example, by noting that a matrix of two-way trunks is symmetric, one can use the same D matrix to store both current and reference trunk matrices. The list of node pairs for sorting can be halved by noting that the trunk increase computed for (i,j) is the same as the trunk increase computed for (j,i) . (These two improvements have

been done in the author's FORTRAN implementation.) Perhaps omission of the sort step will not perturb the results of the algorithm to any significant degree - or perhaps a better sorting algorithm may be used. Omission of the sort step would reduce the time complexity to $O(n^3)$. Perhaps, a better approximation of the internal traffic between node pairs can be used instead of the number of trunks.

SUR

The computational effort of the SUR method may be reduced by experimenting to see if similar results (within tolerance) can be obtained with:

- i) use of a fixed traffic decoupled sets matrix, and
- ii) application of SIZE only for every 2 applications of REROUTE within the same period.

Modification i) would leave out the execution of DECOUP after every run of REROUTE. Modification ii) would halve the number of runs of SIZE.

SUR and Extended Sizing Up

SUR and Extended Sizing Up may be combined to yield an even better multihour engineering algorithm than either one alone. SUR exploits rerouting to reuse spare trunk capacity. Extended Sizing Up sizes up each traffic decoupled set separately to avoid premature engineering of trunks due to the lack of a priori knowledge of traffic demand in a future period. Although SUR will be less effective on a network engineered with Extended Sizing Up (due to tighter trunk allocations and therefore less spare capacity) the combined algorithm should still yield more trunk savings than either one alone.

Extension of SUR to Other Networks

The ideas behind SUR, primarily that of rerouting to make use of spare capacity, may be introduced into the engineering methodology of other types of networks such as private networks and data networks.

7.3 Summary

In this final chapter, we have presented our conclusions on the SUR method and given our recommendations for future work. SUR has proven to be a viable improvement over sizing up for the purpose of multihour dimensioning. Future work in this area may lead to an improved SUR algorithm. Alternatively, the ideas upon which SUR is based may be adapted for use in the engineering of other types of networks, such as private and data networks.

REFERENCES

- [1] Mina, Ramses R., "Introduction to Teletraffic Engineering", Telephony Publishing Corporation, 1974.
- [2] Kuhn, P., BNR-Workshop Traffic Engineering Notes, University of Stuttgart, 1984.
- [3] Rapp, Y., "Planning of Junction Networks in a Multi-Exchange Area", Ericsson Technics No. 1, 1964.
- [4] Beshai, M.E.; Pound, L.A.; Horn, R.W.; "Traffic Data Reduction for Multiple-Hour Network Dimensioning", Networks 83, Conference Publication No. 215, pp.112-118, March 1983.
- [5] Horn, R.W., "A Simple Approach to Dimensioning a Telecommunication Network for Many Hours of Traffic Demand", International Conference on Communications, Denver, Colorado, pp. 67.2.1 - 67.2.5, 1981.
- [6] Wilkinson, R.I., "Nonrandom Traffic Curves and Tables for Engineering and Administrative Purposes", Traffic Studies Center, Bell Telephone Laboratories.
- [7] Wilkinson, R.I., "Theories for Toll Traffic Engineering in the USA", Bell System Technical Journal, Vol. 35, pp.421-514, March 1956.
- [8] Cooper, R.B., "Introduction to Queueing Theory", The Macmillan Company, New York/London, 1972.
- [9] Truitt, C.J., "Traffic Engineering Techniques for Determining Trunk Requirements in Alternate Routing Trunk Networks", Bell System Technical Journal, Vol. 33, pp.277-302, March 1954.
- [10] Syski, R., "Introduction to Congestion Theory in Telephone Systems", Oliver and Boyd Ltd., London, 1960.
- [11] Bear, D., "Principles of Telecommunication - Traffic Engineering", Peregrinus, 1976.
- [12] Freeman, R.L., "Telecommunication System Engineering: Analog and Digital Network Design", Wiley, New York, 1980, ISBN 0-471-02955-6.
- [13] USITA Traffic Committee Bulletin No. 70-2, dated September 1970.

END

2	1	0	2	8	6
---	---	---	---	---	---

FIN