OPTIMAL ROUTING AND DIMENSIONING FOR PRIVATE TELECOMMUNICATION NETWORKS

Kan Jin

A Thesis

in

The Department of Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québéc, Canada

April 1995

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch
Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-01348-0

ABSTRACT

Optimal Routing and Dimensioning for Private Telecommunication Networks

Kan Jin

This thesis investigates some optimal routing and dimensioning problems in private telecommunication networks. In particular, routing and capacity allocation problems for single-service, multiple-service and multipoint circuit-switched networks have been formulated and solved. Attention has been focused on the case that the links of a network are leased from a communication carrier such as the facilities of the public telephone network, and the overflow traffic on each link of the network can be transported by the third party carrier. The optimization problems have been posed as fixed-point problems using node and link flows, which make them easily solvable in real time. The multipoint formulation has been extended to handle the optimal routing problem for multicast packet-switched traffic. The solutions to these problems involve a flow-deviation method. The applicability of the models is demonstrated through a number of numerical examples. Finally, the key features of the approaches are highlighted and suggestions for further research are proposed.

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my thesis supervisor, Dr. J. F. Hayes, and co-supervisor, Dr. M. A. Comeau, for their invaluable assistance and constant guidance throughout this work.

My sincere thanks to Dr. M. Mehmet-Ali for his advice and constructive criticism during the final stage in the preparation of the thesis.

Many professors, fellow graduate students and friends at Concordia University have provided many suggestions and much personal help. My sincere thanks go to all, and I would like to mention specifically the help and advice that I received from Dr. Watson Wu during the early stage of this work.

My heartfelt thanks to my foster parents for their love and constant encouragement, and also I wish to thank my relatives in Montréal for their invaluable help in many ways throughout the course of my studies.

Dedicate to my grandmother, *Lian-Hua*

and my parents, *Xue-yi and Wen-yin*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c_{ij}	capacity of link (i,j)
C_{ij}	cost of link (i,j)
d_{ij}	distance between node i and node j
D_{ij}	the length of link (i,j)
$D_i(j)$	the shortest length from node i to node j
N	number of nodes in the network
$r_{il}(j)$	the fraction of the traffic flow at node i destined for node j routed over link (i,l)
D_j	destination nodes set
$r_{il}(D_j)$	the fraction of the traffic flow at node i destined for a set of nodes D_j routed over link (i,l)
$V_{iD_j}^k$	a set of adjacent nodes to node i on tree k
$R_{iV_{iD_j}^k}(D_j)$	the fraction of the traffic flow at node i destined for a set of nodes D_j routed over branches $(i,l_1),(i,l_2),...,(i,l_k)$, $l_1,l_2,...,l_k \in V_{iD_j}^k$
$R_{iD_j}^n(D_j)$	the fraction of the traffic flow at node i destined for a set of nodes D_j routed over the nth Steiner tree interconnecting i with its destination set D_j
$T_{iD_j}^n$	topology matrix of the nth Steiner tree interconnecting i with destination set D_j
$DD_i^n(D_j)$	the differential cost of the nth Steiner tree interconnecting i with destination set D_j

Greek Symbols

η	scale factor
$\lambda_i(j)$	traffic entering the network at node i destined for node j
$\lambda_i(D_j)$	traffic entering the network at node i destined for a set of nodes D_j
$\Lambda_i(j)$	total traffic at node i destined for node i
$\Lambda_i(D_j)$	total traffic at node i destined for a set of nodes D_j
$\Delta_{il}(j)$	reduction of traffic flow $\Lambda_i(j)$ sent on link (i,l)
$\Delta^n_{iD_j}(D_j)$	the variation of fraction of multipoint calls $\Lambda_i(D_j)$ routed over the nth Steiner tree interconnecting (i, D_j)

Chapter 1

INTRODUCTION

1.1 BACKGROUND

The subject of routing and dimensioning in telecommunication networks has long been recognized as a fascinating problem, and still draws substantial attention.

During the last decade, the rapid changes that have occurred in telecommunications have allowed networks to offer their customers sophisticated service capabilities, which represent new opportunities in business as well as in social activities. Circuit-switched telecommunications are evolving from plain old telephony towards the integrated digital network, which already offers a set of new services. This has increased customers demand to establish a 'private network', which benefits from leased and tariffed telecommunication services, to meet a need for growth in their own businesses. Network dimensioning and traffic routing represent a new challenge for network operators, which can be dealt with thanks to the technological innovations in the telecommunications industry.

1.1.1 Evolution of Traffic Routing

Routing, in its simplest form, has existed since the introduction of the first step-by-step switch. The digits dialed by the subscriber were translated directly into outgoing trunk selections. Each switching center stripped off one or more digits and the connection was progressively established to the called party's line

[1]. However, this form of routing, referred to as *direct routing* in Fig. 1.1, makes inefficient use of transmission facilities and is impractical for toll calls where a significant number of switching centers may need to be traversed.

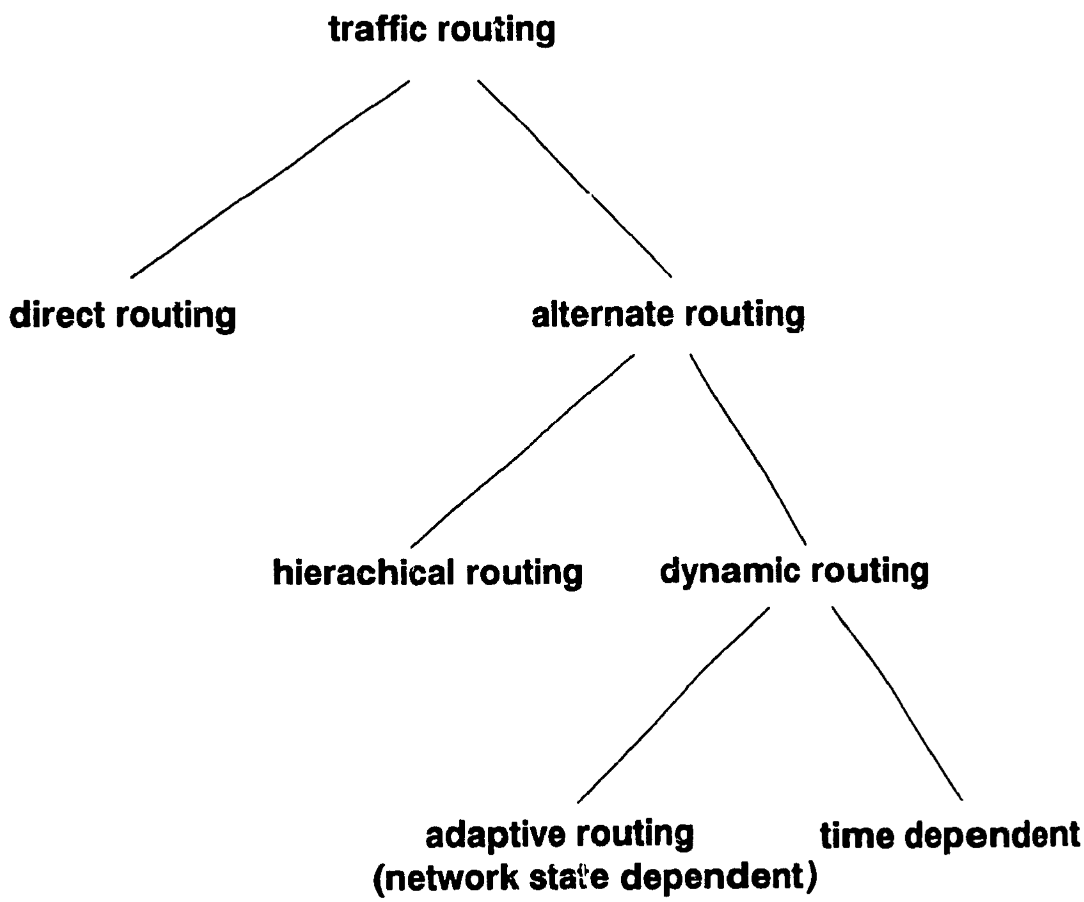


Fig. 1.1 Traffic routing strategy evolution

As soon as common control crossbar switching machines were available in the 1940s and it became technically possible for a switching center to choose a route based on trunk group loading status, the first forms of *alternate routing* were introduced. All switches in a network are of the same type, and serve as a tandem switch. Each switch center would have a number of direct trunks to each of the others and would also have overflow trunks to the tandem. The tandem would connect to every switch serving subscribers. This arrangement enabled significant trunking efficiencies to be achieved.

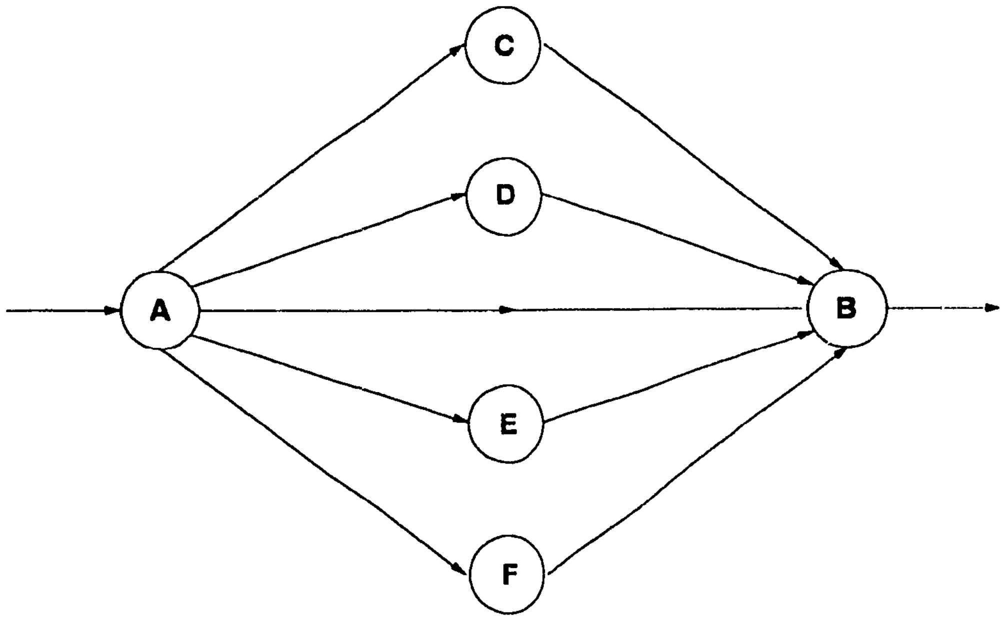
Alternate routing in the network was a little more complex. As there were still technological limitations that required the guarantee of integrity between numbering, signaling and routing, networks were structured into different levels used to concentrate traffic from one region to another. To prevent a call from returning to one of the switching centers along its routing path—the call looping phenomenon—overflowing and the selection of alternative paths were subject to hierarchical rules. In addition, due to limited measurement capabilities in the switches and lack of computing facilities, routing was determined at the design stage and remained fixed under normal conditions until the next design stage. Fixed hierarchical routing was then established [2], and is still used worldwide.

There are however, some limitations to the efficiency gains achievable with hierarchical alternate routing. First of all, it is difficult to predict the effects of noncoincident busy hours for east/west traffic, and noncoincident busy seasons for north/south traffic, which inevitably leads to some over-provisioning. Secondly, there is no way to take advantage of spare capacity, which might be available on high usage routes as a result of off-peak traffic levels; the worst case busy hour always has to be provided for. Besides, the hierarchy is designed on the basis of all equipment being fully operational. The fixed alternates programmed into each switch allow very little adaptability to any significant changes in the traffic patterns. Major failures cause major local congestion. To address this problem, a range of network management controls are provided at the switches (e.g. re-routing within the hierarchy, set code blocks to stop traffic from entering the network for a particular destination, set trunk reservation levels in order to reserve a certain number of trunks for direct traffic only). These are, in general, manually activated by telephone company staff once they become aware of an overload situation. Inevitably there will be time delays while these are being manually implemented, causing some loss of traffic.

With the advent of Stored Program Controlled (SPC) switches, the emergence of common channel signalling systems, and the development of data networks, new processing capabilities allowed for the evolution of traffic routing from fixed hierarchical to flexible techniques referred to as *dynamic routing* in order to improve network efficiency while decreasing network cost. Two dynamic routing strategies have been developed: *time-dependent routing*, where time-variable routing patterns are preplanned according to noncoincidence of predictable traffic peaks; and *state-dependent* (or *adaptive) routing*, where routing is updated on-line according to instantaneous traffic demand and network status information to respond to real-time traffic fluctuations [3].

The principle of routing pattern changes in a time-dependent routing scheme is shown in Fig. 1.2. The direct path and two-link alternate paths are used for carrying traffic between origin and destination switches. The time-varying routing capability allows prespecified routing patterns to change as frequently as every consistent time period, such as one hour, to respond to forecast traffic profiles. A crankback technique is used to identify downstream blocking on the second link of each two-link alternate path, and a call blocked at the transit switching center will be re-routed. Crankback is very efficient at small traffic loads, but may impose additional load at the origin switch when the traffic grows. This is due to the fact that some extra processing capability is needed to attempt the next route of the overflow traffic. The introduction of the time-dependent routing concept, which led to the successful implementation of Dynamic Nonhierarchical Routing (DNHR) in AT & T's long distance network [4], has resulted in the improvement of network efficiency and reduction of network costs. Though network cost are considered similar in time-dependent and adaptive routing, differences in their principles imply specific implementation.

An adaptive routing scheme tracks, in real time, spare capacity in the network. To consider the routing problem, let us reduce the network to its essential structural features—a set of nodes (switches) interconnected by links (trunks), with each link having a capacity in terms of number of circuits, each of which can carry a call. A call originating at node A and destined for node B requires, for its completion, a single-link or multiple-link path between nodes A and B, with at least one idle circuit on each link of the path at the time of call origination. The selected route, consisting of one circuit on each link of the path, is then occupied for the duration of the call, at the end of which the circuit or circuits become available for the connection of other calls. Thus, calls arrive at random and are routed on suitable paths whenever possible, rendering those paths unavailable to other calls for the duration of the routed calls. The traffic produces random var-



Time Period	A to B Route Pattern
1	AB, ACB, ADB
2	AB, AEB, AFB
3	ADB, AFB, AB

Fig. 1.2 Routing pattern changes in time-dependent routing

iations in the link occupancies in the network. The aim of a routing scheme is to influence, to the extent possible, these random variations in the pattern of link occupancies so as to minimize network blocking (the probability of an arriving call finding no suitable idle path for its connection). It becomes evident that the routing problem can be characterized as a Markov decision process, i.e., as the probabilistic control of a system. The first rigorous approach toward adaptive routing was presented by V. E. Benes [5] in 1966, which treated the traffic routing problem as a Markov decision process with an average-reward criterion. However, the number of states in the Markov chain associated with the routing problem, even for small networks, is astronomically large and defies the sort of computation required in the algorithms for optimum solution.

A number of investigations were directed using insights of Markov decision theory to obtain useful routing algorithms [6][7]. The exact determination of cost for optimum routing is a hopeless task for practical networks, since the number of equations that must be solved is equal to the size of the state space. So an area of continuing interest is trying to derive the relative costs from some reasonable initial routing schemes without having to solve the state space equations.

Three generic systems have been proposed so far for adaptive routing in telecommunication networks: centralized, distributed and isolated. In a centralized scheme there is a central entity that places the calls in the most efficient way in order to optimize the global performance of the network. In a distributed scheme, a decentralized routing algorithm is performed based on global information about the network state. Conversely, in isolated schemes there are local entities that try to optimize their local performance.

Centralized optimization routing models were first considered in packet-switched networks, due to the flexibility of the routing for such networks. Fratta, Gerla and Kleinrock [8] formulated the centralized optimization static routing problem as a convex programming problem and introduced the flow deviation algorithm, which is an adaptation of the Frank-Wolfe algorithm used to solve convex multicommodity formulations. Gallager [9] followed the same trend suggested by Fratta et al, but for quasi-static routing problems using distributed computation. Also in packet-switching, Gavish and Hantler [10] proposed a centralized non-linear combinatorial optimization problem to tackle the problem of route selection in a computer network while Mason [11] described analytic models for isolated adaptive routing and computed off-line the centralized optimization routing patterns.

In the area of circuit-switching, the centralized and decentralized adaptive routing approaches with some particularly simple algorithms are exemplified by several systems, such as the Dynamically Controlled Routing/High Performance Routing (DCR/HPR) system developed by Bell Northern Research [12], the TSMR system proposed by AT&T Bell Lab for the DNHR network [13], the System to Test Adaptive Routing (STAR) implemented by CNET in the Paris network operated by France Telecom [14], the Dynamic Routing - 5 minutes (DR-5) system proposed by Bellcore for the metropolitan networks operated by the Bell Operating Companies (BOCs) [15], the Dynamic Alternative Routing (DAR) system developed by British Telecom for their trunk network [16] and the State and Time-dependent Routing (STR) system used by Nippon Telegraph and Telephone (NTT) [17].

Many powerful routing schemes, either centralized, distributed or isolated, have been proposed and analyzed but they have not been explicitly considered for real implementation because of complexity. Instead, they have represented a theoretical background to support simpler solutions. Mason and Girard [18] presented an unifying framework for a number of routing techniques and evaluated performance models by comparing them with exact Markov chain analysis. More recently, Kelly [19][20], Dziong et al [21] and Girard [3] presented a centralized approach based on revenue maximization and showed how the existence of shadow prices associated with the routes of the network could be used as a basis for a decentralized adaptive routing scheme. The new insights gained are leading to the formulation of efficient routing schemes.

1.1.2 Challenge of Multimedia and Multipoint Communication

Increasing demand for telecommunication services of voice, text, data and image as well as the new capacities provided by the revolutionary advances in communications technology presented a new set of complexities to the network planners and designers.

In the new context with new service mixes and higher network resource capacities, there is a need to find the new economies of scale and equilibria to match user need and technology capabilities. The degree of expansion of new service is to a great extent a function of the new techno-economic equilibria among services demand, quality of service and network resource provisioning. The planning function in a complex environment is more than ever viewed as a "decision-making" process, which requires updating-extension of the classical procedures of network design and routing optimization.

The focus of some research efforts has been on methods that deal with the problem of routing calls for different services (such as voice, video, etc.) in a network [22][23]. As in the case of single-service networks, these issues are concerned about the application of Markov decision theory. Similarly, a major difficulty in applying Markov decision theory to solve multirate routing problem is the computational complexity.

While the techniques of multirate communications are being explored, the issues involved in the design of multipoint communication networks became a hot topic. The motivation for multipoint communication comes from the recognition that there is a wide class of applications that require it. The most obvious one is the distribution of entertainment programs, either audio or video. Also there are many other services for which multipoint communication is important. A few of these are listed below.

- *Wire services.* News services like the Associated Press and Reuters distribute news reports from their bureaus to newspapers and radio stations throughout the world. A general multipoint connection would allow efficient transmission of this information from a moderate number of sources to a much large number of receivers.

- *Multi-point conferences.* These services are currently handled by routing through a central point. They could be provided more efficiently and flexibly using a general multipoint connection.

- *Video lecture.* An important special case of a multi-person conference is a video lecture in which one speaker addresses a large audience, with provision for audience members to ask questions. Such a service could play an important role in education if it could be provided conveniently and inexpensively.

- *LAN interconnection.* Most large companies have local computer networks such as Ethernet at multiple locations. An extremely attractive service for them would be a multipoint connection that makes their geographically distributed LANs appear to be one large network. Such a service would allow them to treat local and remote computers uniformly, allowing them to take advantage of the large base of networking software developed for LANs. The networking model offered by a multipoint-LAN is an attractive one for distributed operating systems and database applications. The extension of this model to geographically distributed networks would be very popular.

These examples indicate the possible range of services that multipoint connections can support. There is a much larger number of potential services that might become attractive if flexible and economical techniques were widely deployed, including switching system design, connection management, routing and overload control, etc.

There are relatively few methods dealing with the multipoint routing problem. It is typically treated as the Steiner tree problem in graph theory and is known to be NP-complete. This means that the existence of efficient algorithms for finding the optimal route is unlikely. Consequently, one is forced to turn to algorithms that may produce sub-optimal solutions, but which can be expected to work acceptably well in practice.

It turns out that for the Steiner tree problem, there are several known approximation algorithms that provide a good starting point for work on multipoint routing [24][25][26][27][28][29]. However, these algorithms are able to route only a single multipoint call each time. To compute the routes for multiple-multipoint calls, these algorithms have to be applied sequentially to each call, which is obviously inefficient. Recently, Noronha Jr. and Tobagi [30] formulated the problem of routing multicast streams as an integer programming problem for data networks. But for large networks, the optimum solution that they presented was not practical. Its main use is as benchmark for other multicast routing algorithms. To this extent, finding a good practical algorithm remains an art.

1.1.3 Network Dimensioning

The network dimensioning problem is to provide, at minimum cost, enough capacity to ensure that performance is acceptable, where the performance could be an overall average blocking criterion, a requirement on grade of service in circuit-switched networks or the average delay constraints in data networks. The problem has been addressed by many researchers; surveys on dimensioning of telecommunication networks are provided in [31][32][33][34][35]. A variety of mathematical formulations, exact and heuristic procedures for solving it, has been suggested.

In general, the network dimensioning process is decoupled from the routing process. However, routing must be taken into account when dimensioning a network, since the routing method determines how effectively the link capacities are utilized. The simultaneous optimization of routing and capacity allocation has been considered as a hard combinatorial problem up to now.

1.2 SCOPE OF THE THESIS

The objective of the thesis is to investigate some optimal routing and capacity allocation problems for single-service, multiple-service and multipoint circuit-switched private networks. It should be noted that any routing policy, that optimizes the management of calls on the network, has an impact not only on network performance but on network dimensioning as well. The purpose of the thesis is to introduce an optimization model that would integrate several pertinent factors to be considered and that would be versatile and easily solvable for real-sized networks. It gives rise to an elegant formulation in which routing process and dimensioning process are combined together. In the following five chapters we present a method based on the flow deviation approach to solve the routing and dimensioning problems in circuit-switched networks, and extend it to handle the optimal routing problem for multicast packet-switched traffic.

The thesis is organized as follows:

In Chapter 2, we pose the optimization of routing and capacity allocation for point-to-point voice-carrying circuit-switched networks as a fixed-point problem using link and node flows. Implicit in the model is the assumption that accepted traffic will be carried along primary routes belonging to the network and that overflow traffic due to call blocking on primary routes can be carried by a third party on a point-to-point basis at a cost proportional to the carried load. A composite cost function is constructed, which allocates a minimum-cost capacity over each link of a network as a function of its link load distribution, taking into consideration the cost of the overflow traffic. A flow-deviation algorithm is applied to solve this optimization problem. The applicability of the model will be demonstrated through numerical examples.

Integrated point-to-point multirate traffic is then incorporated into the optimization problem in Chapter 3. We consider a circuit-switched network that supports a variety of traffic classes with different traffic characteristics (bandwidth requirement, call arrival rate and call holding time). It is assumed that the third party carrier can also offer the higher-rate circuit-switched capacity between all pairs of points. A multivariate composite cost function is constructed to allocate appropriate capacity over each link as a function of its link load distribution. Then the flow-deviation method for solving the optimization problem in voice-carrying circuit-switched networks is extended to solve the optimization problems in multiple-service circuit-switched networks. The main steps involved in solving the problem is described in this chapter. A practical optimal network is computed and presented as an example.

In Chapter 4, a multipoint routing and dimensioning problem in circuit-switched networks is formulated. Multipoint calls supporting services such as voice and video conferencing require circuits interconnecting all nodes in a subset of network nodes. Under the assumption that the third party needs only to provide point-to-point capacity for the network to support multipoint services, the multipoint optimization problem is posed in a similar framework. Two varieties of solution heuristics for the optimization are developed. The first one is actually an extension of the flow-deviation method for point-to-point traffic and employs heuristics to solve the problem approximately. The second one is a modification of an approach suggested by Bertsekas and Gallager [24] for the broadcast problem in data networks. The performance and run times of two iterative algorithms are compared through some examples.

The multipoint formulation for circuit-switched networks is modified into a form applicable to the multicast routing problem for packet-switched networks in Chapter 5. We will discuss our optimization problem formulations and solution heuristics. The computational power of algorithms will be demonstrated through numerical evaluation results.

Finally, the conclusion and recommendation for further work are presented in Chapter 6.

Chapter 2

OPTIMAL ROUTING AND CAPACITY ALLOCATION FOR POINT-TO-POINT CIRCUIT-SWITCHED NETWORKS

2.1 INTRODUCTION

Routing and dimensioning problems for point-to-point circuit-switched networks have been examined in the literature by Kelly [19][20], Dziong et al [21] and Girard [3] lately. According to these formulations, the network is made up of switches (nodes) interconnected by links and there is an origin-destination (O-D) demand matrix that reflects the traffic needs of the network for a given period of time. The links have a capacity in terms of the number of circuits. Every origin-destination pair of nodes constitutes a different type of call, so, in mathematical terms, it constitutes a different commodity. The demand will be assigned to the links of the network so that some cost or performance function is optimized and the capacity constraints of the network are respected. A performance measure commonly used in these models is the global probability of blocking in the network, which in most cases is determined by Erlang's B formula. This function is non-convex with respect to the traffic variable, and therefore the use of this formula in the objective function of an optimization problem leads to a non-convex problem, which is very difficult to solve due to the capacity constraints linking different commodities.

One technique to solve these problems consists in eliminating the capacity constraints and adding, instead, a penalty term to the objective function. In this way, the problem becomes a nonlinear problem. This was exactly the approach proposed by Fratta, Gerla and Kleinrock in their work done on packet-switched networks. In their case, the capacity constraints were replaced by a convex delay penalty function.

In circuit-switched networks, there is no notion of delay, but rather the notion of blocking. Kelly [19] proposed the expected operating revenue of calls as a penalty function. This not only allows the capacity constraints to be dropped, thus converting a difficult problem into an easier one, but also measures a factor that one would naturally want to maximize in a network. In this approach, the rate of return from the network is the sum, over all routes, of the carried traffic multiplied by the "revenue" of that traffic. The revenue can be arbitrary, but if they are identically one, then the return is just the total carried traffic. A call offered to a route will generate a net expected revenue. The routing offers more traffic to routes with a high net expected revenue. The approach associates a "shadow price" with each link, which is the derivative of the return with respect to the link capacity. The derivative information provided by the shadow price can be used in a hill-climbing algorithm to dimension the network at minimum cost [36].

Solving the routing and dimensioning problem with the above model is very time consuming. Dziong et al [21] and Girard [3] simplified the model under various assumptions.

The purpose of this chapter is to introduce a new type of routing and dimensioning model, closer to reality than formulations mentioned above. We consider the option that lines between any two nodes or points in the network are leased from a communication carrier. The capacity of a leased line must be selected from a finite number of choices, such as DS0, DS1, DS2, DS3 and DS4. The traffic between any two nodes of the network has a "peaked" profile. There is no need for the private network to be able to carry the peak demand using its own resources alone. Instead it is permissible, and far more economic, to 'overflow' the peak demand to the third party such as the public network. We assume that a third party carrier is available to carry all overflow traffic due to blocked calls on the leased lines at a cost proportional to the amount of overflow. No calls are lost in this model. The model assumes that a route for a given call can be composed of some links belonging to the network and others belonging to a third party. In this way, overflow is measured on individual links only (not on paths), and the network may be viewed as a fully-connected topology with infinite link capacities.

The rest of this chapter is organized as follows. Section 2.2 contains a formal statement of the problem. In section 2.3, we provide a detailed treatment of routing procedures in networks, focusing on the shortest path algorithm. Section 2.4 contains some numerical examples and conclusions are contained in section 2.5.

2.2 GENERAL FORMULATION OF THE PROBLEM

2.2.1 The Network

The network interconnecting N nodes will be represented initially by a completely-connected virtual topology. We consider all links to be point-to-point and undirected. Each link is characterized by: (i) Capacity—link bandwidth, in terms of number of circuits; (ii) Cost—monetary cost of using the link, in \$/month. Formally, a network with N nodes is denoted by $G(A, W)$, where A is the $N \times N$ topology matrix, its (i, j) element is 1 if link (i, j) exists and 0 otherwise, W is the matrix of link parameters, containing the capacity and the cost of each link.

2.2.2 The Traffic

The voice call arrivals for the different node pairs are taken to be independent Poisson streams. All call holding times are assumed to have exponential distribution with the same mean value $1/\mu$.

Allowing for calls between all $N(N-1)$ distinct origin-destination pairs, stationary rates $\lambda_i(j)$ are given for the exogenous arrivals of voice calls at node i that are destined for node j. Rather than describing the routing of calls along explicitly specified paths connecting each O-D pair, it is more convenient for our purposes to use a set of node flow variables $\Lambda_i(j)$ giving the total arrival rate of calls at node i that are destined for node j, together with a set of routing variables or routing tables $r_{il}(j)$ for each node i, specifying the fraction of all calls at node i and destined for node j that are to be routed along link (i, l). Flow conservation for each type of traffic is expressed through the traffic equation

$$\Lambda_i(j) = \lambda_i(j) + \sum_l \Lambda_l(j) r_{li}(j), \quad i, j \in \{1, ..., N\} \tag{2.1}$$

where $0 \le r_{il}(j) \le 1$ and $\sum_l r_{il}(j) = 1$.

In a practical circuit-switched network, the capacity of a line is typically equal in both directions, although the one-way link is also adopted in a network

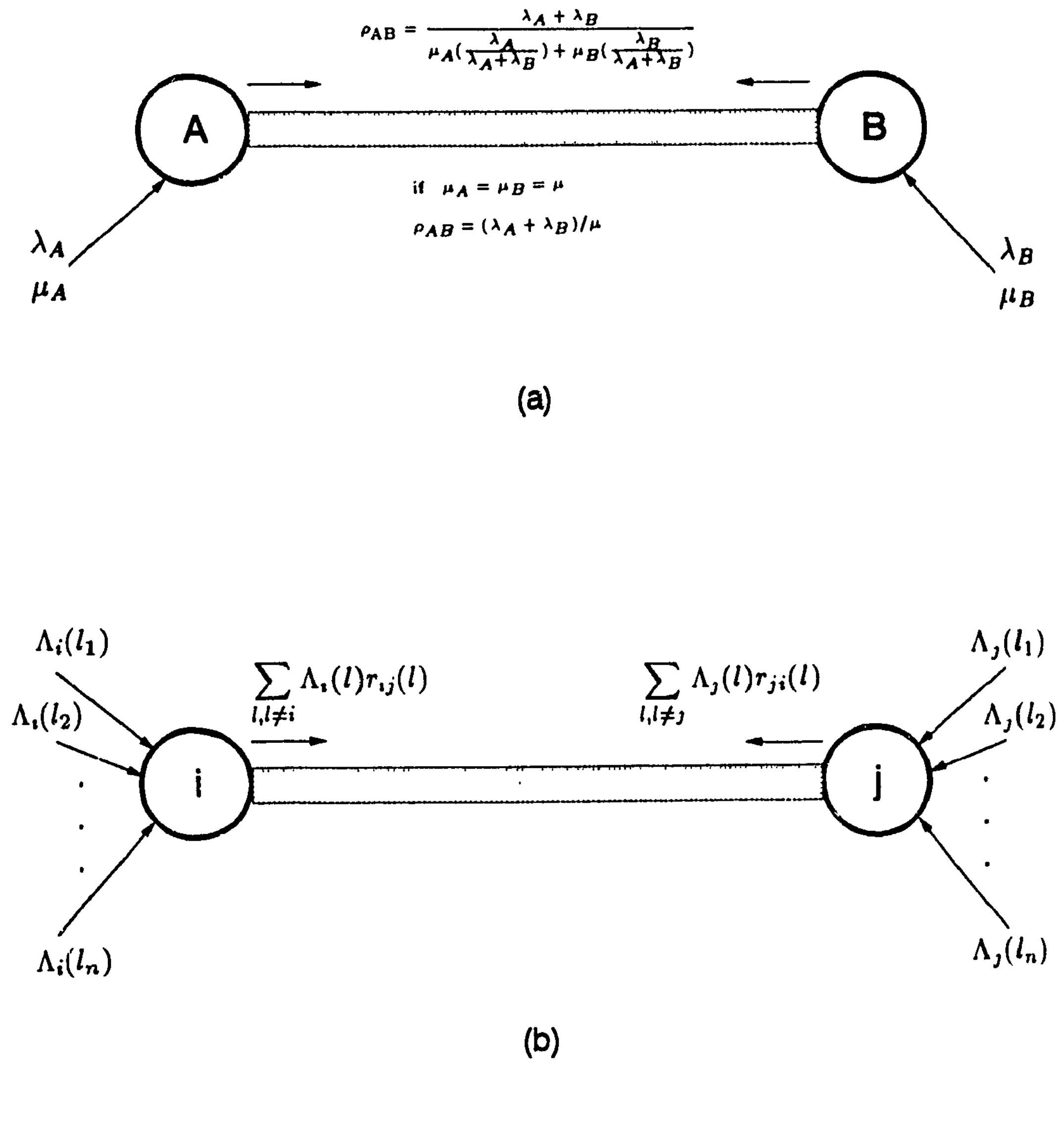


Fig. 2.1 Traffic load on a two-way link

for some cases [37]. We only consider the case of two-way links. Fig. 2.1(a) illustrates an example of how the traffic load on a two-way link is calculated [38]. As in Fig. 2.1(b), the total traffic load on link (i,j) is computed in a duplex fashion according to

$$\rho_{ij} = [\sum_{l,l \neq i} \Lambda_i(l) r_{ij}(l) + \sum_{l,l \neq j} \Lambda_j(l) r_{ji}(l)]/\mu \tag{2.2}$$

2.2.3 The Cost Function

The links of backbone networks for long-haul communication are usually leased from a communication carrier such as the facilities of the public telephone network. The standard of some communication carriers (used in North America, also used in Japan) is shown in Table 2.1.

Table 2.1 Capacity of some communication carriers

Line type	Number of circuits	Data rate (Mbps)
$DS0$	1	0.064
$DS1$	24	1.544
$DS1C$	48	3.152
$DS2$	96	6.312
$DS3$	672	44.736
$DS3C$	1344	90.432
$DS4$	4032	274.176

The capacity c_{ij} allocated to link (i,j) depends on the level of the link load ρ_{ij}. For each value of ρ_{ij}, we can compute the cost of operating a link (i,j) for each of the available capacities $\{nDS0, nDS1, nDS1C, nDS2, nDS3, nDS3C, nDS4\}$ and any combinations thereof. It is assumed that the overflow on the link will be transported by a third party carrier such as the public telephone network. Operating a link (i,j) entails a substantial fixed monthly fee for leasing capacity c_{ij}, as well as the cost incurred by having all overflow traffic o_{ij} on link (i,j) by the third party carrier.

The leasing fee includes distance charge, equipments charge, tax and miscellaneous services charge. We list all these rates in Table 2.2, Table 2.3 and Table 2.4.

Let:

- K total number of circuits
- n number of DS0, DS1, DS2, DS3 or DS4 lines
- β_1 distance rate for leasing a line
- β_2 base charge for leasing a line
- β_3 IX (interexchange) link connectors charge (connection to PSTN (public switched telephone network))
- β_4 access charge (connect LEC (local exchange carriers) lines to switches)
- β_5 box (contain small voltmeters connected to lines) charge

β_6 contribution (tax) charge

β_7 miscellaneous services charge

γ discount rate (50% for long-term subscribers. In the rest of the thesis, we use γ=0.5 in all cases.)

The cost of leasing a line is

$$C_1(c_{ij}, d_{ij}) = C_0(c_{ij}, d_{ij}) + \beta_6 + \beta_7 K \tag{2.3}$$

where

$$C_0(c_{ij}, d_{ij}) = \begin{cases} \gamma[(\beta_1 d_{ij} + \beta_2 + 2\beta_3)n + 2\beta_5] + 2\beta_4 n, & \text{for DS0} \\ [\gamma(\beta_1 d_{ij} + \beta_2 + 2\beta_3) + 2\beta_4 + 2\beta_5]n, & \text{for DS1 or DS2, DS3, DS4} \end{cases}$$

Based on empirical data amassed in the North American PSTN, the average time of a subscribers's line being busy is 9 hours per day and 21 days per month. Each month, the total overflow traffic o_{ij} on link (i,j) is measured by

$$\begin{aligned} o_{ij} &= 9 \times 21 \times \rho_{ij} \times B(c_{ij}, \rho_{ij}) \\ &= 189\rho_{ij} B(c_{ij}, \rho_{ij}) \end{aligned} \tag{2.4}$$

where ρ_{ij} is expressed in Erlangs, which define the average traffic density during a one-hour period. $B(c_{ij}, \rho_{ij})$ determines the blocking probability on link (i,j) by Erlang's B formula.

Then the total cost of operating link (i,j) when assigned capacity c_{ij} can be expressed as

$$\begin{aligned} C_{ij}(h_{ij}, d_{ij}, c_{ij}, o_{ij}) &= C_1(c_{ij}, d_{ij}) + \alpha_{ij} o_{ij} \\ &= C_1(c_{ij}, d_{ij}) + 189\alpha_{ij}\rho_{ij} B(c_{ij}, \rho_{ij}) \end{aligned} \tag{2.5}$$

where α_{ij} is the cost in \$/erlang, its rate is shown in Table 2.5.

Table 2.2 Distance rate for leasing a line

Distance (miles)	Capacity	Charge per mile($)	Base charge ($)
1 – 25	DS0	17.00	0.00
	DS1	200.00	0.00
	DS2	580.00	0.00
	DS3	1800.00	0.00
	DS4	5205.00	0.00
26 – 50	DS0	12.00	130.00
	DS1	135.00	1555.00
	DS2	340.00	4400.00
	DS3	1190.00	13560.00
	DS4	3509.00	39000.00
51 – 100	DS0	8.00	350.00
	DS1	91.00	3620.00
	DS2	263.00	10510.00
	DS3	815.00	32425.00
	DS4	2450.50	95120.00
101 – 200	DS0	4.10	652.00
	DS1	49.00	7810.00
	DS2	143.00	22756.00

Table 2.2 Distance rate for leasing a line (cont'd)

Distance (miles)	Capacity	Charge per mile($)	Base charge ($)
101 - 200	DS3	433.00	70250.00
	DS4	1301.00	211000.00
201 - 500	DS0	2.20	1040.00
	DS1	26.00	12370.00
	DS2	74.00	36100.00
	DS3	227.00	112500.00
	DS4	680.00	348000.00
501 - 1000	DS0	1.60	1350.00
	DS1	19.00	16000.00
	DS2	53.00	46600.00
	DS3	163.00	144000.00
	DS4	495.50	435000.00
over 1000	DS0	0.60	2350.00
	DS1	7.00	28000.00
	DS2	18.00	81600.00
	DS3	55.00	251700.00
	DS4	170.00	761500.00

Table 2.3 Equipment charges

Capacity	Connector charge ($)	Access charge ($)	Box charge ($)
DS0	35.00	15.00	350.00
DS1	305.00	160.00	40.00
DS2	1220.00	640.00	160.00
DS3	8540.00	4480.00	1120.00
DS4	51240.00	26880.00	6720.00

Table 2.4 Contribution charge

Number of circuit	Contribution charge ($)
1 - 3	22.00
4 - 6	72.00
7 - 9	112.00
10 - 14	132.00
15 - 19	152.00
20 - 29	172.00
30 - 39	192.00
40 - 49	202.00
50 - 74	212.00
75 - 100	232.00
over 100	240.00

Table 2.5 Rate of carrying the overflow traffic

Distance (miles)	Rates ($/erlang)
1 - 60	18.00
61 - 400	21.00
401 - 1000	27.00
1001 - 2000	28.80
over 2000	31.2

The link cost with assigned capacities $2DS0$ and $8DS0$ is depicted in Fig. 2.2, where distance is 75 miles.

We choose c^*_{ij} among all available capacities so as to minimize $C_{ij}(\rho_{ij}, d_{ij}, c_{ij}, o_{ij})$ for each value of the link load ρ_{ij}. In this way, we get the cost function

$$C_{ij}(\rho_{ij}, d_{ij}, c^*_{ij}, o_{ij}) = \begin{cases} \underbrace{\min}_{c_{ij} \in \{DS0,\ldots,DS4\}} \{C_{ij}(\rho_{ij}, d_{ij}, c_{ij}, o_{ij})\} & i \neq j \\ 0 & i = j \end{cases} \quad (2.6)$$

where $\rho_{i,j} \in [0, \infty)$

The cost function assigns a least-cost solution independently for each link, the capacity is accommodated in a cost-effective fashion. The cost function of a link with distance $d_{ij} = 75$ miles is shown in Fig. 2.3. Note that it is not worthwhile to lease some capacities when the traffic load is very light. A link with zero capacity does not represent a link that in effect does not exist.

2.2.4 Problem Formulation

<u>Problem Definition</u>

Formally, the problem can be stated as: "Given a network $G(A, W)$ and stationary traffic rates for call arrivals between arbitrary pairs of nodes, minimize the total cost of the network over all choices of routing variables. The corresponding link load (recall that link load is a function of arrival rates and routing variables), in turn, dimensions the network $G(A, W)$ by uniquely specifying a least-cost capacity for each link." Thus, routing and network dimensioning are intrinsically combined.

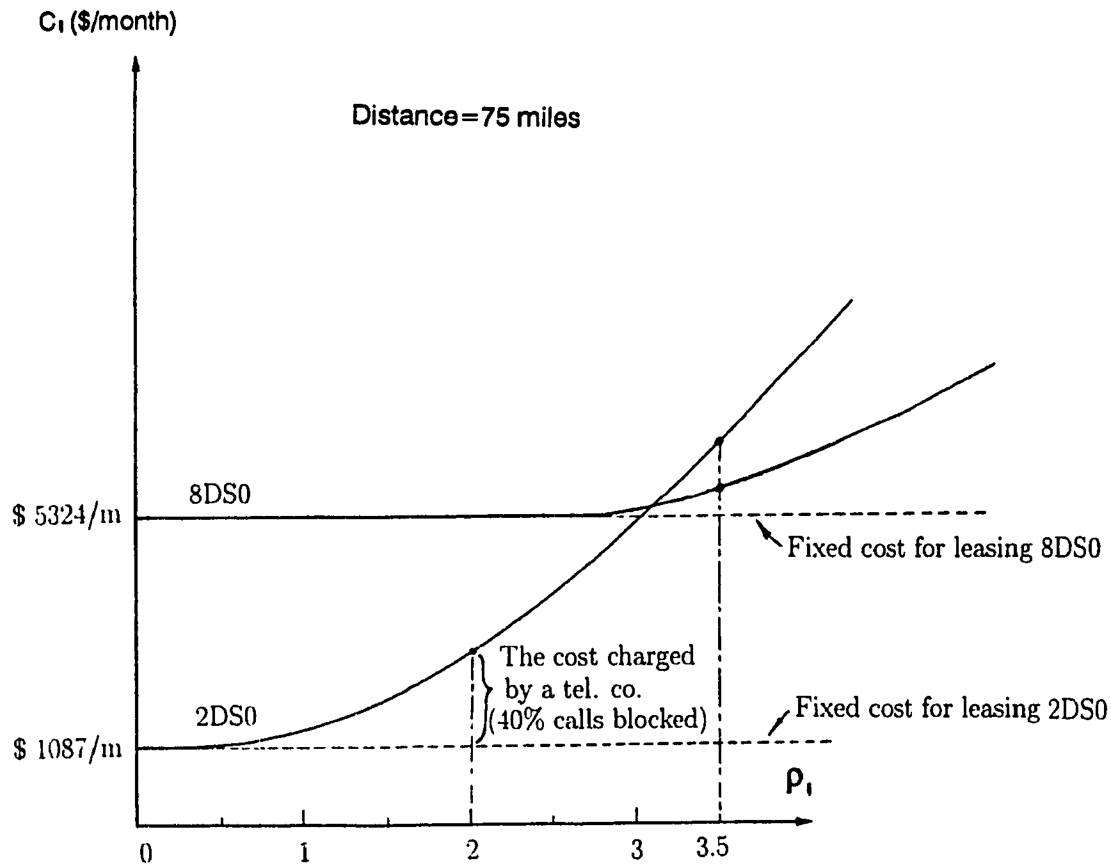


Fig. 2.2 Link cost with specific assigned capacities

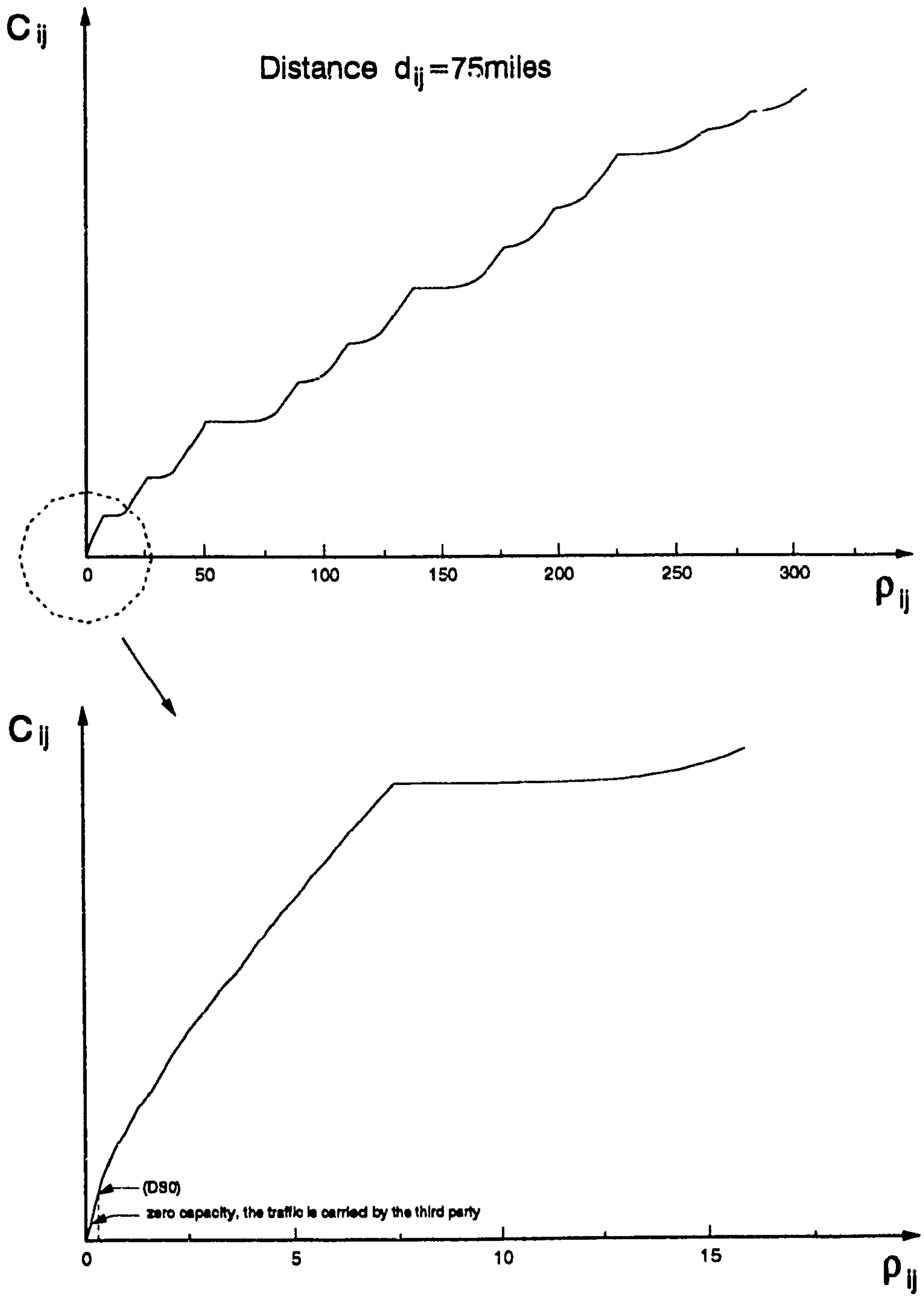


Fig. 2.3 Link cost function

Formulation

Definitions:

N	Number of nodes in the network
$\lambda[\lambda_{ij}]$	$N \times N$ external traffic matrix
$\mathbf{d}[d_{ij}]$	$N \times N$ matrix, where d_{ij} is the distance between node i and node j in miles
$\mathbf{o}[o_{ij}]$	$N \times N$ matrix, where o_{ij} is the overflow traffic on link (i,j)
$\alpha[\alpha_{ij}]$	$N \times N$ matrix, where α_{ij} is the cost in \$/erlang for carrying the overflow traffic o_{ij}
$\mathbf{c}^*[c^*_{ij}]$	Capacity matrix, where c^*_{ij} is the least-cost capacity allocated to link (i,j)
$\mathbf{C}[C_{ij}]$	Cost matrix, where C_{ij} is the cost of link (i,j)
R	The routing variable set $\{r_{il}(j)\}$

GIVEN: N, λ, $\mathbf{d}$, α

MINIMIZE:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} \tag{2.7}$$

where C_{ij} is given by Eq. (2.6)

$$C_{ij}(\rho_{ij}, d_{ij}, c^{*}_{ij}, o_{ij}) = \begin{cases} \underbrace{\min}_{c_{ij} \in \{DS0,\dots,DS4\}} \{C_{ij}(\rho_{ij}, d_{ij}, c_{ij}, o_{ij})\} & i \neq j \\ 0 & i = j \end{cases}$$

WITH RESPECT TO : R

UNDER CONSTRAINTS:

1. Flow conservation: satisfy Eq. (2.1)

$$\Lambda_i(j) = \lambda_i(j) + \sum_l \Lambda_l(j) r_{li}(j), \quad i,j \in \{1,\dots,N\}$$

2. No traffic at a destination loops back into the network:

$$r_{jl}(j) = 0, \qquad j,l \in \{1,2,\dots,N\} \tag{2.8}$$

2.3 THE ALGORITHM

The algorithm to solve the above optimization problem is similar in form to the one presented by Gallager in [9], for minimum delay routing in packet-switched networks. Centralized versions are given here, although, the algorithms are also suitable for distributed computation.

The algorithm consists of three procedures as follows.

Initialization

Initially, a fully connected network is assumed. For N nodes, there will be $N(N-1)/2$ links in the network. All traffic is routed over the direct links connecting sources to destinations by defining

$$r_{il}(j) = \begin{cases} 1 & if\ l = j \\ 0 & otherwise \end{cases} \tag{2.9}$$

An example is given in Fig. 2.4, only the traffic destined from node 1 to 4 goes on link $(1, 4)$.

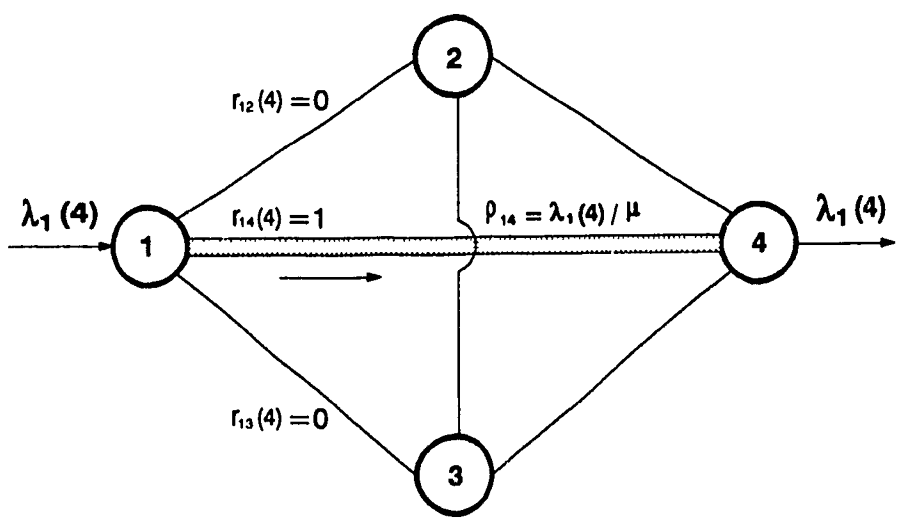


Fig. 2.4 Initial traffic flow allocation

Updating

In each iteration thereafter, we update the node and link flows using (2.1) (2.2), respectively. The differential cost of each link is then computed as

$$D_{ij} = \frac{\partial C_{ij}(\rho_{ij}, d_{ij}, c^*_{ij}, o_{ij})}{\partial \rho_{ij}} \tag{2.10}$$

Cycling through the O-D pairs in successive iterations, each iteration processes a single O-D pair i, j. We use the values computed from (2.10) as the length of a link and determine the shortest paths from all nodes in the network to node j.

Let: $P_l(j)$ denote a path from node l to node j

$SP_l(j)$ denote the shortest path from node l to node j

D_{lj}= the length of link (l, j)

$D_l(j)$= $\min \sum_{(k,m)\in P_l(j)} D_{km}$, the length of the shortest path $SP_l(j)$

$\Delta_{il}(j)$= the amount of reduction of $r_{il}(j)$

Then, for each link (i, l) we compute

$$\Delta_{il}(j) = \min\left\{ r_{il}(j), \eta(1 - \frac{D_i(j)}{D_{il} + D_l(j)}) \right\}, \quad \eta \in [0, 1] \tag{2.11}$$

and update the routing variables at node i for destination j according to

$$r_{il}(j) = \begin{cases} r_{il}(j) - \Delta_{il}(j), & \text{if link } (i,l) \text{ is not on } SP_i(j) \\ r_{il}(j) + \sum_{(i,l)\notin SP_i(j)} \Delta_{il}(j), & \text{if link } (i,l) \text{ is on } SP_i(j) \end{cases} \tag{2.12}$$

Routing variables updates are essential to ensure that the traffic flows adapt to varying link load patterns. The algorithm reduces the fraction of traffic sent on the links on non-shortest path and increases the fraction on the link on the shortest path.

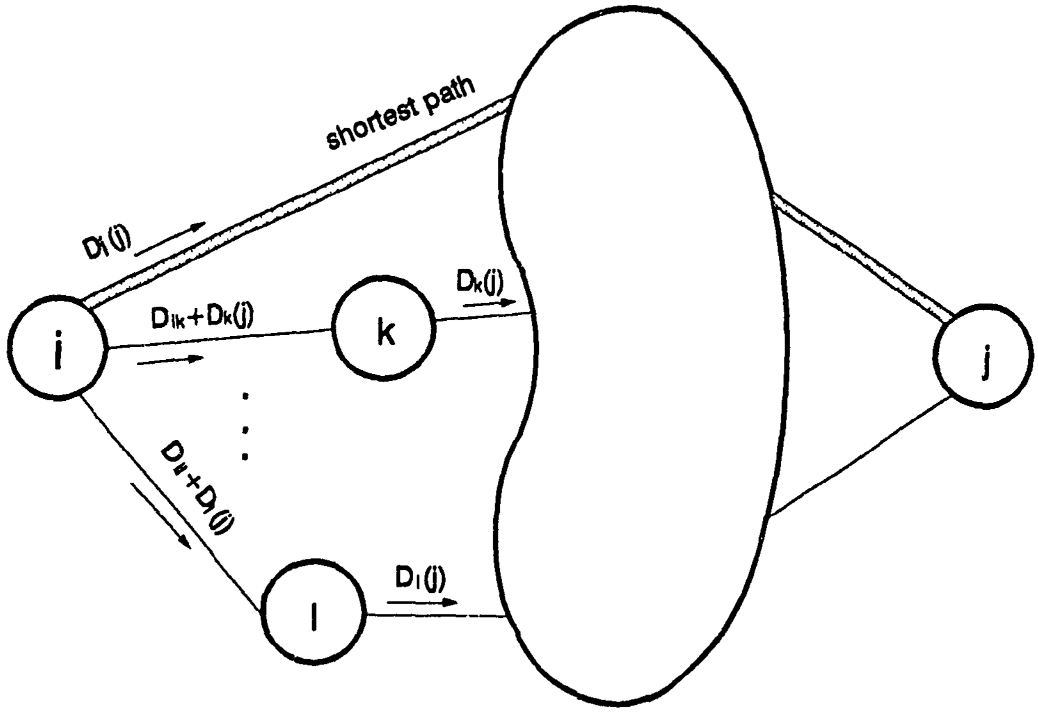


Fig. 2.5 Basic idea of $\Delta_{il}(j)$.

$D_{ik} + D_k(j), ..., D_{il} + D_l(j)$ are the length of non-shortest path from node i to j, while $D_i(j)$ is the length of the shortest path from i to j. $\frac{[D_{ik}+D_k(j)]-D_i(j)}{D_{ik}+D_k(j)}, ..., \frac{[D_{il}+D_l(j)]-D_i(j)}{D_{il}+D_l(j)}$ are relative differences between the length of the non-shortest and the length of the shortest path.

The amount of reduction, $\Delta_{il}(j)$, is proportional to the relative difference between the length of the path from node i to j using link (i, l), which is not on the $SP_i(j)$, and using the link on the $SP_i(j)$, with the restriction that $r_{il}(j)$ can not be negative. Fig. 2.5 illustrates the basic idea of the reduction. Note that as the length of non-shortest path approaches the length of the shortest path, the reductions get smaller. The amount of reduction also depends on the scale factor η. The value of η will affect the speed of convergence.

Termination

The algorithm terminates after the kth iteration if

$$[r_{il}(j)]^{k+1} = [r_{il}(j)]^{k}, \qquad \forall i, j, l \in \{1, ..., N\} \tag{2.13}$$

This algorithm is executed in an iterative fashion. The processing of iteration is expressed in Fig. 2.6.

2.4 NUMERICAL RESULTS

In this section, we give some numerical results to illustrate performance of the algorithm described in section 2.3.

Recall from section 2.2, the cost function (Eq.(2.6)) allocates a minimum-cost capacity over each link of a network as a function of its link load distribution, taking into consideration the cost of overflow traffic. Traffic overflowing on each link is measured with Erlang's B formula. Its calculation is involved in each iteration of the algorithm. Thus, the efficacy of the algorithm depends, in large part, on improved computations of the Erlang B function and its derivatives. Several authors have studied and presented many mathematical methods and

```
Initialization;
WHILE (termination criteria are not met) {
   FOR j = 1 to N {
      FOR i = 1 to N {
         /* update all node and link flows by (2.1), (2.2) */
         /* calculate all link lengths by (2.10)
             and determine SP_i(j), for all i ∈ {1,...,N} */
         /* update routing variables r_il(j), for all l ∈ {1,...,N} */
         /* test termination conditions */
      }
   }
}
```

Fig. 2.6 The procedure of the algorithm algorithms for this problem [39][40][41][42]. We are interested in the approach that enabled the construction of convenient programs in [42].

Erlang's B formula is usually stated in the form

$$B(K,\rho) = \frac{\frac{\rho^K}{K!}}{\sum_{j=0}^{K} \frac{\rho^j}{j!}} \tag{2.14}$$

or, equivalently,

$$\begin{aligned} B(K,\rho)^{-1} &= \sum_{j=0}^{K} \frac{K!}{j!} \rho^{-(K-j)} \\ &= \sum_{j=K}^{0} \frac{K!}{(K-j)!} \rho^{-j} \end{aligned} \tag{2.15}$$

where K is the number of circuits.

In terms of the descending factorial, $K^{(j)}$, defined by

$$K^{(0)} = 1, K^{(j)} = K(K-1)\cdots(K-j+1), \quad j \geq 1 \tag{2.16}$$

$B(K,\rho)^{-1}$ may also be written as

$$B(K,\rho)^{-1} = \sum_{j=0}^{K} K^{(j)} \rho^{(-j)} \tag{2.17}$$

From the Eulerian integral

$$\rho^{-j} = \rho \int_0^\infty e^{-\rho y} \frac{y^j}{j!} dy \tag{2.18}$$

one has

$$B(K,\rho)^{-1} = \rho \int_0^\infty e^{-\rho y} \sum_{j=0}^{K} \frac{K!}{j!} y^j dy \tag{2.19}$$

and hence

$$B(K,\rho)^{-1} = \rho \int_0^\infty e^{-\rho y}(1+y)^K dy \qquad (2.20)$$

since the above integral has meaning for nonintegral K, the general definition of $B(X,\rho)$, in which $X > 0$ is unrestricted and $\rho > 0$ is

$$B(X,\rho)^{-1} = \rho \int_0^\infty e^{-\rho y}(1+y)^X dy \qquad (2.21)$$

This relates $B(X,\rho)$ to the incomplete gamma function, namely,

$$\begin{aligned} B(X,\rho)^{-1} &= \rho^{-X} e^{\rho} \int_\rho^\infty e^{-y} y^X dy \\ &= \rho^{-X} e^{\rho} \Gamma(X+1,\rho) \end{aligned} \qquad (2.22)$$

integration by parts applied to (2.21) yields

$$B(X,\rho)^{-1} = 1 + X \int_0^\infty e^{-\rho y}(1+y)^{X-1} dy \qquad (2.23)$$

hence,

$$B(X,\rho)^{-1} = \frac{X}{\rho} B(X-1,\rho)^{-1} + 1 \qquad (2.24)$$

This is an efficient recursion for the successive computation of $B(X,\rho)^{-1}$. For integral values of X, the initial value $B(0,\rho) = 1$ is convenient.

Now we study the performance of the algorithm via several practical networks through experiments. The performance metric in which we are interested is the total cost of a network.

A five-node network

The original traffic input matrix and distance matrix of a five-node network are specified in table 2.6 and table 2.7, respectively. All parameters related to the cost of a link in the network are already shown in Table 2.2, Table 2.3, Table 2.4 and Table 2.5. In the experiments, we set $1/\mu = 3.5$ mins, which corresponds to the mean duration of a voice call.

Table 2.6 Original traffic matrix of a five-node network (calls/min)

$\lambda_i(j)$	1	2	3	4	5
1	—	1.98	0.53	1.12	0.30
2	2.00	—	3.885	8.50	24.33
3	0.68	3.98	—	0.72	10.54
4	1.76	11.34	1.60	—	7.45
5	0.46	17.82	9.11	8.14	—

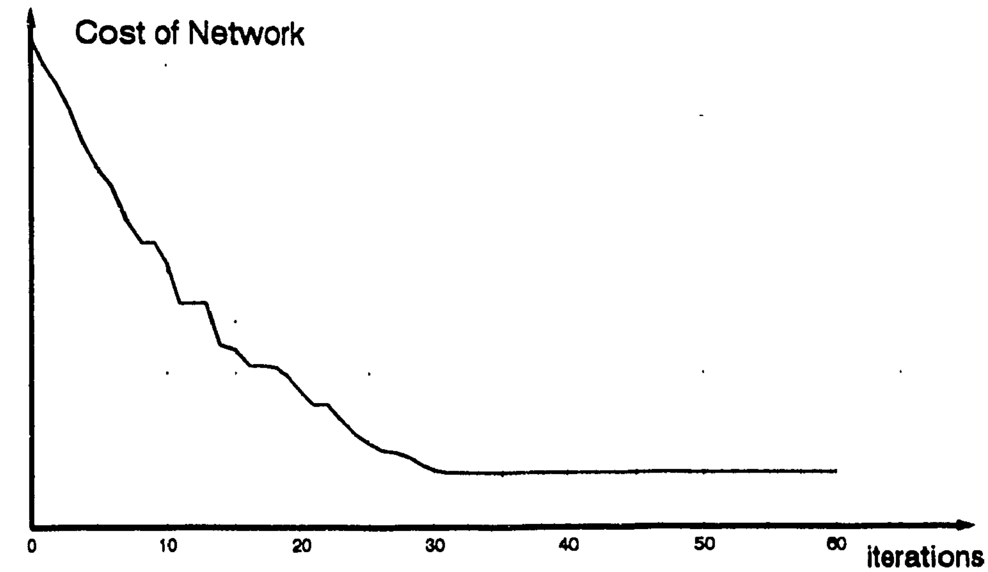
Table 2.7 Distance matrix of a five-node network (miles)

$d_i(j)$	1	2	3	4	5
1	—	2807.5	2862.2	2881.9	3000.6
2		—	57.0	249.4	337.0
3			—	276.0	345.0
4				—	118.7
5					—

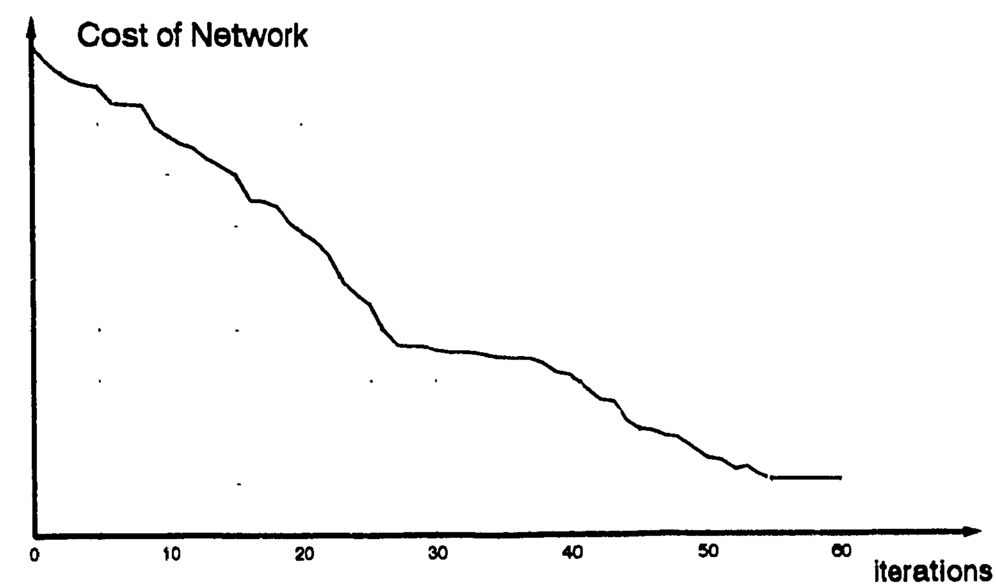
Initially, all traffic is allocated to the directed links connecting sources and destinations. The initial cost of the 5-node network is \$347.64/month.

In general, the convergence point depends on the choice of the scale factor η. There are several ways to choose the value of η. One possibility is to keep η as a constant. With this choice, a crucial question has to do with the magnitude of this constant. Experimental work is necessary to determine a practical value for η to guarantee converging to an optimal point. Another straightforward way is choosing $\eta \in (0, 1]$ to achieve a largest decrease in the objective function in each iteration, if this η does not exist, then simply set η to zero. Thus, the routing variables remain unchanged in the iteration, the convergence of the algorithm is guaranteed. It is also possible to choose η by a simple form of line search. First, start with $\eta = 1$, update routing variables and evaluate the corresponding total cost of the network C ($C = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}(\rho_{ij}, d_{ij}, c_{ij}^{*}, \alpha_{ij})$), and if no reduction is obtained over C, successively reduce η until a cost reduction is obtained or until $\eta = 0$.

Fig. 2.7(a) and (b) illustrate the network cost over iterations for different selection methods of the value of η. The converged results for each case are shown in Fig. 2.8 and 2.9 respectively. For both cases, the selection of η involves repeated calculation of node flows, link flows and network cost in each iteration. In spite of longer iterations, choosing η by line search results in less CPU time, while the operation of determining an η to achieve a minimum network cost at each iteration will substantially increase computational time. This phenomenon is very typical for a large network. Seen in Fig. 2.8 and Fig. 2.9, the two different schemes of η selection do not make any significant difference in the network cost and topology.

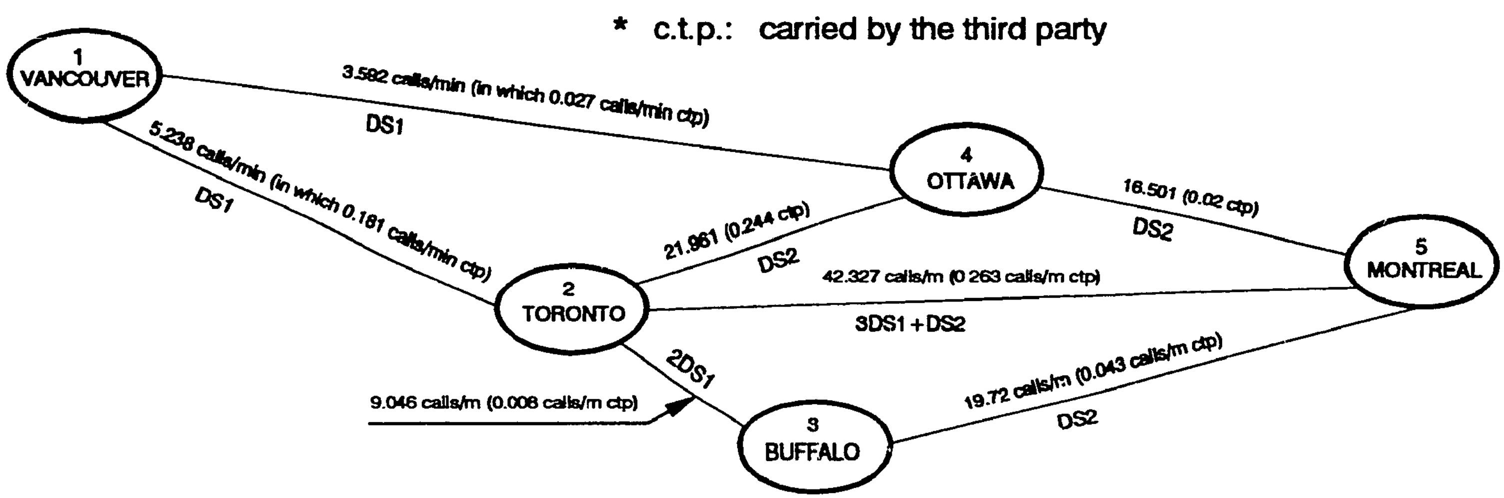


(a) η is chosen to achieve the largest decrease in the objective function



(b) η is chosen by line search

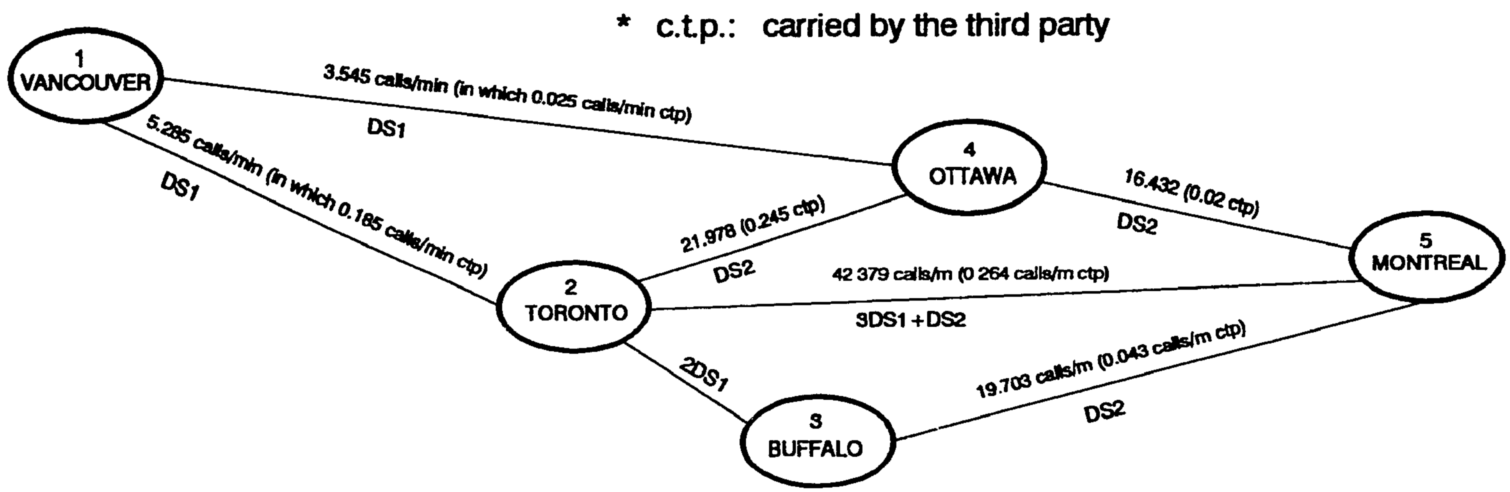
Fig. 2.7 Effect of η on the speed of convergence



$r_{12}(2)=1$	$r_{21}(1)=1$	$r_{32}(1)=1$	$r_{41}(1)=1$	$r_{54}(1)=0.9544$
$r_{12}(3)=1$	$r_{23}(3)=1$	$r_{32}(2)=1$	$r_{42}(2)=1$	$r_{52}(1)=0.0456$
$r_{14}(4)=1$	$r_{24}(4)=1$	$r_{32}(4)=0.7770$	$r_{42}(3)=0.9761$	$r_{52}(2)=0.9982$
$r_{12}(5)=0.0920$	$r_{25}(5)=1$	$r_{35}(4)=0.2230$	$r_{45}(3)=0.0239$	$r_{54}(2)=0.0018$
$r_{14}(5)=0.9080$		$r_{35}(5)=1$	$r_{45}(5)=1$	$r_{53}(3)=1$
				$r_{54}(4)=1$

Network Cost= \$ 305.27 k/month (Initial Cost= \$ 347.62 k/month)

Fig. 2.8 Example 1 of a 5-node network



$r_{12}(2)=1$	$r_{21}(1)=1$	$r_{32}(1)=1$	$r_{41}(1)=1$	$r_{54}(1)=0.8410$
$r_{12}(3)=1$	$r_{23}(3)=1$	$r_{32}(2)=1$	$r_{42}(2)=1$	$r_{52}(1)=0.1590$
$r_{14}(4)=1$	$r_{24}(4)=1$	$r_{32}(4)=0.8001$	$r_{42}(3)=1$	$r_{52}(2)=1$
$r_{12}(5)=0.1074$	$r_{25}(5)=1$	$r_{35}(4)=0.1999$	$r_{45}(5)=1$	$r_{53}(3)=1$
$r_{14}(5)=0.8926$		$r_{35}(5)=1$		$r_{54}(4)=1$

Network Cost= $ 305.29 k/month (Initial Cost= $ 347.62 k/month)

Fig. 2.9 Example 2 of a 5-node network

According to our experiment, choosing η by line search usually works well in practice.

A ten-node network

Table 2.8 Original traffic matrix of a 10-node network (calls/min)

$\lambda_i(j)$	1	2	3	4	5	6	7	8	9	10
1	–	1.0	1.2	1.3	1.5	2.0	0.5	1.8	1.0	0.8
2	1.2	–	0.75	1.1	0.85	1.8	0.4	0.9	0.9	1.4
3	1.1	0.65	–	2.5	1.7	4.0	2.8	4.5	1.2	0.78
4	1.5	1.2	3.0	–	0.7	1.0	0.6	0.6	1.5	2.4
5	1.6	1.0	1.5	0.5	–	8.0	10.4	1.4	0.8	3.6
6	2.2	1.6	4.4	0.8	7.5	–	4.8	10.5	24.3	2.1
7	0.4	0.6	2.6	0.5	10.0	5.2	–	3.1	10.5	1.0
8	2.0	0.8	4.8	0.6	1.1	9.8	2.6	–	2.2	1.8
9	1.1	0.9	1.0	1.7	1.2	21.8	9.2	2.8	–	4.3
10	0.7	1.3	1.1	2.2	3.2	1.75	1.45	2.3	5.0	–

Table 2.9 Distance matrix of a 10-node network (miles)

d_{ij}	1	2	3	4	5	6	7	8	9	10
1	–	777.5	660.6	1048.0	1132.0	2807.5	2872.0	2882.0	3000.0	3170
2		–	187.0	330.0	849.0	2159.4	2219.0	2217.0	2353.0	2522.0
3			–	388.0	835.0	2146.0	2206.0	2220.0	2340.0	2509.0
4				–	518.0	1829.0	1889.0	1903.0	2022.0	2192.0
5					–	1312.0	1372.0	1386.0	1505.0	1673.0
6						–	57.0	249.0	337.0	506.0
7							–	228.0	345.0	495.0
8								–	118.0	287.0
9									–	168.0
10										–

Table 2.8 and Table 2.9 demonstrate the call arrivals and distance between nodes of a 10-node intercity network. Using the proposed algorithm, the corresponding optimal value of network cost is equal to \$1033.144K/month (initial network cost=\$1173.015K/month), 11.9% down from the initial cost. The network topology is shown in Fig. 2.10. The routing fractions over links, the link flows and capacities assignments are listed in Table 2.10 and Table 2.11, respectively. As can be seen, the network are not densely connected. The average number of paths per point-to-point pair was 1.79, the longest path consists of 5 links.

A twenty-node network

The traffic arrivals matrix and distance matrix of a twenty-node network are presented in Table 2.12 and Table 2.13. The final results are given in Fig. 2.11. We see that the capacities assigned on several links are zero, the traffic takes advantage of the third party carrier. The link flows and capacities assignments are summarized in Table 2.14, and routing fractions are listed in Table 2.15. The cost of network is $5251.795K/month, a 10.7% decrease from the initial case (initial network cost=$5883.348K/month).

2.5 SUMMARY

In this chapter, we have proposed a new type of routing and dimensioning model for circuit-switched networks. The preliminary research has assumed that a third party carrier is available to handle overflow traffic on any link. Based on this notion, a composite cost function is constructed, which allocates a minimum-cost capacity over each link as a function of its link load. The interesting point to note is that the definition of the objective function has been extended to simplify the routing and dimensioning procedures, they are combined intrinsically, in contrast to conventional techniques. It has been found through numerical experiments that convergence to a near-optimum is possible by properly choosing parameters and that the algorithm is efficient.

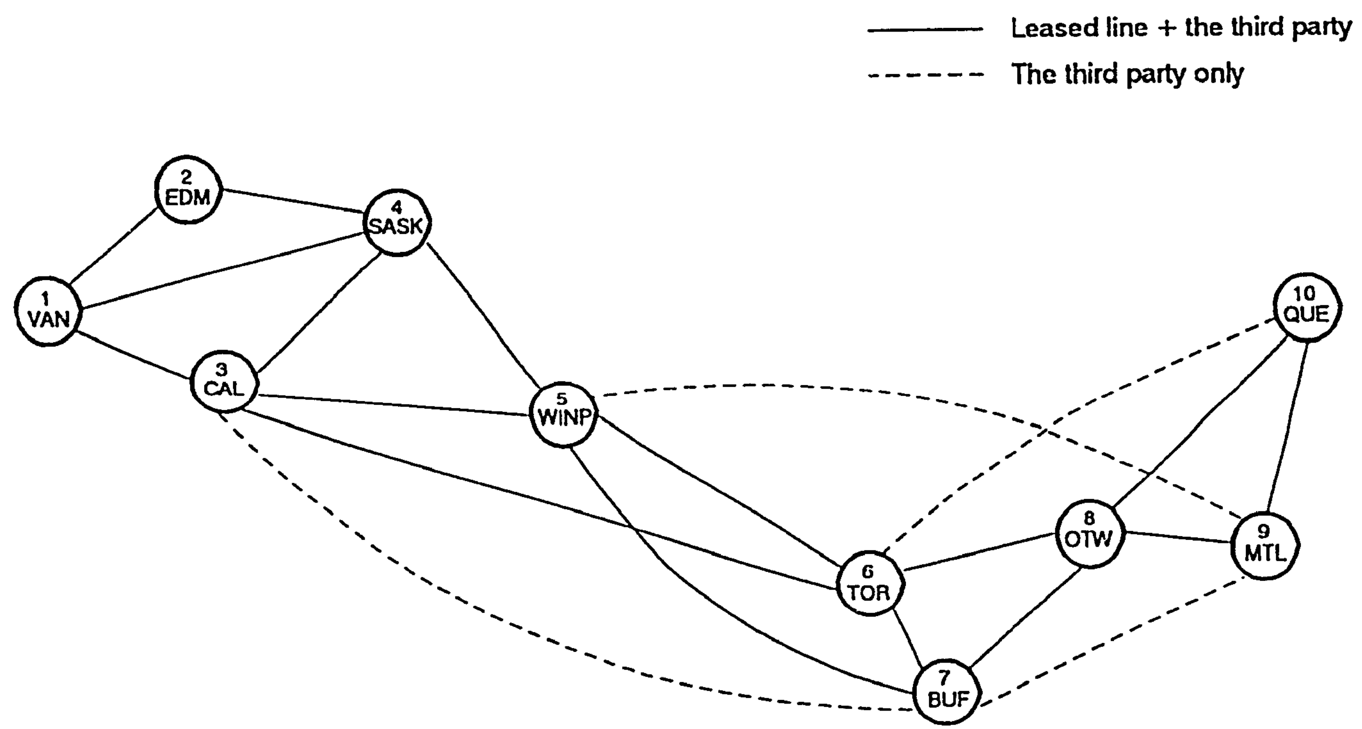


Fig. 2.10 Final topology of the 10-node network

Table 2.10 Routing fractions of a 10-node network

Source	Destination	Link	Fraction	Source	Destination	Link	Fraction
	2	(1,2)	1 0000			(3,5)	0.4314
	3	(1,3)	1 0000	3	9	(3,6)	0 5669
	4	(1,4)	1 0000			(3,7)	0.0017
1	5	(1,3)	1 0000		10	(3,6)	1.0000
	6	(1,3)	1 0000		1	(4,1)	1.0000
	7	(1,3)	1 0000		2	(4,2)	1.0000
	8	(1,3)	1 0000		3	(4,3)	1.0000
	9	(1,3)	1 0000		5	(4,5)	1.0000
	10	(1,3)	1 0000	4	6	(4,5)	1.0000
	1	(2,1)	1 0000		7	(4,5)	1 0000
	3	(2,1)	0 5719		8	(4,5)	1.0000
		(2,4)	0 4281		9	(4,5)	1.0000
	4	(2,4)	1.0000		10	(4,5)	1.0000
2	5	(2,4)	1 0000		1	(5,3)	1.0000
	6	(2,4)	1 0OCO		2	(5,4)	1.0000
	7	(2,4)	1 0000		3	(5,3)	1.0000
	8	(2,4)	1 0000		4	(5,4)	1.0000
	9	(2,4)	1.0000	5	6	(5,6)	1.0000
	10	(2,4)	1.0000		7	(5,6)	0.1789
	1	(3,1)	1 0000			(5,7)	0.8211
	2	(3,1)	0.7932		8	(5,6)	1.0000
		(3,4)	0.2068			(5,7)	0.0002
	4	(3,4)	1.0000		9	(5,8)	0.9997
	5	(3,5)	1 0000			(5,9)	0.0001
	6	(3,5)	0.3223		10	(5,6)	0.9990
		(3,6)	0 6777			(5,9)	0.0010
3		(3,5)	0 1234		1	(6,3)	0.6432
	7	(3,6)	0.8755			(6,5)	0.3568
		(3,7)	0.0011	6	2	(6,5)	1.0000
	8	(3,5)	0 3528		3	(6,3)	0 8919
		(3,6)	0 6472			(6,5)	0.1081

Table 2.10 Routing fractions of a 10-node network (cont'd)

Source	Destination	Link	Fraction	Source	Destination	Link	Fraction
6	4	(6,5)	1 0000	8	8	(8,6)	1 0000
	5	(6,5)	1 0000		9	(8,6)	1 0000
	7	(6,7)	1 0000		10	(8,6)	1 0000
	8	(6,8)	1 0000	9	1	(9,8)	1.0000
	9	(6,9)	1.0000		2	(9,8)	1.0000
	10	(6,8)	0.9993		3	(9,7)	0 0014
		(6,10)	0 0007			(9,8)	0 9986
7	1	(7,3)	0 0102		4	(9,8)	1 0000
		(7,5)	0 6205		5	(9,5)	0 0100
		(7,6)	0 3693			(9,8)	0 9900
	2	(7,5)	1.0000		6	(9,8)	1 0000
	3	(7,3)	0 0025		7	(9,7)	0 0054
		(7,5)	0.7519			(9,8)	0 9946
		(7,6)	0 2456		8	(9,8)	1 0000
	4	(7,5)	0 6298		10	(9,10)	1 0000
		(7,6)	0 3702	10	1	(10,6)	0 0105
	5	(7,5)	1 0000			(10,8)	0 9895
	6	(7,6)	1.0000		2	(10,8)	1 0000
	8	(7,8)	1.0000		3	(10,6)	0 0009
	9	(7,8)	0.9899			(10,8)	0.9979
		(7,9)	0.0101			(10,9)	0 0012
	10	(7,8)	0 9999		4	(10,8)	0 9885
		(7,9)	0 0001			(10,9)	0 0115
8	1	(8,6)	1 0000		5	(10,8)	0.9998
	2	(8,6)	1.0000			(10,9)	0 0002
	3	(8,6)	1 0000		6	(10,8)	1 0000
	4	(8,6)	1.0000		7	(10,8)	1 0000
	5	(8,6)	1.0000		8	(10,8)	1 0000
	6	(8,6)	1 0000		9	(10,9)	1 0000

Table 2.11 Link flow and capacity allocation of a 10-node network

Link	Capacity (leased line)	Traffic flow (calls/min)	
		Leased line	The third party
(1,2)	1 DS1	3.144	0.001
(1,3)	1 DS2	18.843	0.002
(1,4)	1 DS1	2.800	0.0
(2,4)	2 DS1	12.585	2.299
(3,4)	1 DS1	5.289	0.345
(3,5)	3 DS1	17.136	0.365
(3,6)	1 DS1 + 1 DS2	28.312	0.149
(3,7)	—	—	0.018
(4,5)	1 DS2	24.491	1.059
(5,6)	1 DS1 + 2 DS2	56.776	1.290
(5,7)	1 DS2	22.607	0.208
(5,9)	—	—	0.020
(6,7)	3 DS1	15.882	0.091
(6,8)	7 DS1 + 3 DS2	117.277	0.211
(6,10)	—	—	0.016
(7,8)	1 DS1 + 1 DS2	27.619	0.080
(7,9)	—	—	0.157
(8,9)	1 DS1 + 3 DS2	81.279	0.679
(8,10)	1 DS1 + 1 DS2	27.74	0.089
(9,10)	2 DS1	9.312	0.023

Table 2.12 Original traffic matrix of a 20-node network

$\lambda_i(j)$	1	2	3	4	4	6	7	8	9	10
1	—	1.2	1.3	1.4	1.5	2.0	8.0	2.0	1.8	4.0
2	1.2	—	0.75	0.9	1.4	0.85	1.4	2.8	3.5	1.6
3	1.3	1.0	—	3.1	2.7	1.2	3.3	0.9	1.8	2.8
4	1.5	1.4	3.0	—	4.3	1.2	9.4	1.0	5.4	3.2
5	1.6	0.8	2.0	4.0	—	2.3	1.1	2.6	3.5	7.3
6	2.1	1.3	1.0	1.0	2.0	—	1.8	2.4	1.0	1.4
7	8.2	1.0	3.0	9.0	0.8	1.6	—	1.4	1.8	2.9
8	1.8	2.0	1.0	0.8	3.0	2.0	1.0	—	3.7	3.9
9	2.2	3.0	2.0	5.0	4.0	0.95	1.5	3.0	—	2.2
10	3.7	2.0	2.4	3.0	8.0	1.2	2.5	4.2	2.5	—
11	1.7	4.0	0.75	1.0	2.2	1.8	3.0	1.8	2.0	4.8
12	2.8	10.0	7.0	2.0	1.3	2.2	4.0	3.9	3.2	1.0
13	0.95	3.5	1.0	1.0	2.0	2.4	4.5	1.0	3.1	3.0
14	1.0	3.0	1.4	2.2	3.5	3.9	4.0	2.3	1.2	2.0
15	1.2	3.0	0.8	2.0	1.7	0.9	5.0	3.0	2.5	0.8
16	1.0	4.0	1.9	0.75	1.5	3.0	6.0	3.5	1.0	2.4
17	1.9	1.0	2.1	2.0	1.4	3.4	4.3	1.4	1.0	3.3
18	3.6	0.85	3.1	1.8	4.0	3.0	1.5	0.9	1.5	5.2
19	2.1	6.0	2.7	1.9	4.2	3.25	1.5	2.0	1.7	4.4
20	1.25	2.5	1.0	1.1	1.0	4.4	2.0	1.4	1.0	3.0

Table 2.12 Original traffic matrix of a 20-node network (cont'd)

$\lambda_i(j)$	11	12	13	14	15	16	17	18	19	20
1	1.5	2.6	0.85	0.65	0.9	1.1	1.7	3.4	1.9	0.95
2	4.2	9.7	2.1	2.8	3.2	3.7	0.8	0.5	6.7	2.2
3	0.5	7.7	1.3	1.0	0.85	1.9	2.1	2.3	2.0	0.4
4	1.1	2.4	0.7	1.8	2.1	0.45	1.7	1.5	1.0	0.85
5	1.4	0.78	2.1	3.0	1.6	1.4	1.3	3.3	3.9	1.5
6	2.1	2.4	2.8	4.2	1.1	3.4	3.8	3.2	3.5	4.1
7	3.2	3.8	4.1	3.8	5.7	6.9	4.0	2.0	1.9	2.2
8	2.1	4.2	0.8	2.7	3.5	4.0	1.6	0.4	2.2	1.0
9	1.7	3.0	2.9	0.85	2.1	0.7	0.6	1.3	1.5	1.2
10	5.2	1.1	3.2	1.8	0.4	2.7	3.1	4.9	4.2	2.8
11	—	0.7	2.2	2.8	3.4	3.6	1.9	4.1	4.5	4.2
12	0.9	—	2.8	4.3	0.9	1.1	4.9	0.7	8.6	6.6
13	2.0	3.0	—	2.2	1.2	1.6	1.4	2.8	1.8	2.9
14	2.5	4.0	2.0	—	4.3	5.7	2.7	1.7	0.5	2.2
15	3.0	1.1	1.0	4.5	—	0.75	1.25	2.97	1.8	3.1
16	4.0	1.5	2.0	6.0	1.15	—	0.88	1.24	2.5	1.0
17	2.2	5.2	1.0	3.0	0.99	1.3	—	1.6	3.4	2.0
18	4.3	1.2	3.0	2.0	3.12	1.5	2.1	—	1.4	1.8
19	4.0	8.0	2.0	0.9	2.2	2.4	3.0	1.6	—	2.5
20	4.0	8.1	3.0	2.0	3.2	1.3	1.7	2.1	3.05	—

Table 2.13 Distance matrix of a 20-node network

d_{ij}	1	2	3	4	4	6	7	8	9	10
1	—	1057.0	1244.0	1677.0	1822.0	2232.0	2812.0	3833.0	4331.0	4295.0
2		—	299.0	620.0	764.0	1336.0	2026.0	2776.0	3273.0	3238.0
3			—	528.0	785.0	1357.0	2082.0	2767.0	4241.0	3259.0
4				—	257.0	829.0	1555.0	2269.0	2766.0	2731.0
5					—	571.0	1297.0	2012.0	2509.0	2474.0
6						—	726.0	1440.0	1938.0	1902.0
7							—	648.0	1107.0	1073.0
8								—	459.0	425.0
9									—	414.0
10										—

Table 2.13 Distance matrix of a 20-node network (cont'd)

d_{ij}	11	12	13	14	15	16	17	18	19	20
1	4492.0	4559.0	4611.0	5034.0	4801.0	5021.0	5071.0	5414.0	5399.0	6050.0
2	3434.0	3502.0	3553.0	3977.0	3743.0	3964.0	4014.0	4319.0	999.0	4973.0
3	3455.0	3523.0	3547.0	3998.0	3764.0	3985.0	4053.0	4340.0	4363.0	5013.0
4	2927.0	2995.0	3046.0	3454.0	3236.0	3547.0	3507.0	3813.0	3835.0	4485.0
5	2670.0	2737.0	2789.0	3212.0	2979.0	3199.0	3249.0	3555.0	3578.0	4228.0
6	2099.0	2166.0	2218.0	2641.0	2408.0	2623.0	2678.0	2984.0	2953.0	3656.0
7	1493.0	1447.0	1669.0	1721.0	1859.0	1828.0	2129.0	1931.0	2188.0	3108.0
8	830.0	784.0	1230.0	4883.0	1370.0	1180.0	1640.0	1283.0	1540.0	2618.0
9	809.0	763.0	1218.0	769.0	1318.0	907.0	1590.0	1035.0	1344.0	2371.0
10	369.0	322.0	768.0	809.0	908.0	916.0	1178.0	1019.0	1112.0	2157.0
11	—	68.0	399.0	900.0	539.0	880.0	809.0	885.0	908.0	1788.0
12		—	467.0	832.0	609.0	813.0	877.0	818.0	840.0	1856.0
13			—	1014.0	190.0	772.0	460.0	777.0	735.0	1439.0
14				—	932.0	212.0	1178.0	327.0	686.0	1640.0
15					—	745.0	270.0	613.0	546.0	1249.0
16						-	961.0	160.0	473.0	1427.0
17							—	864.0	629.0	982.0
18								—	332.0	1278.0
19									—	921.0
20										—

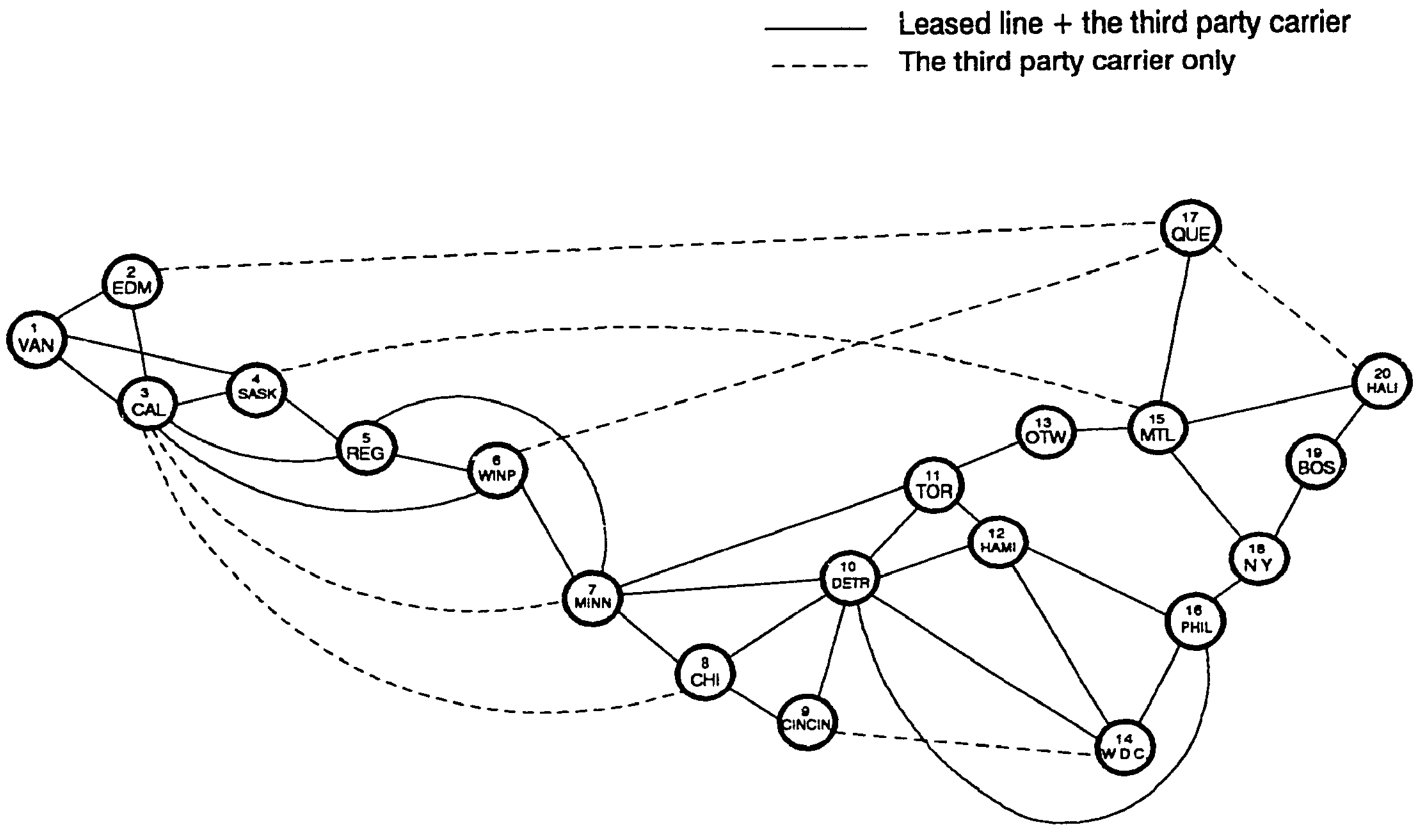


Fig. 2.11 Final topology of the 20-node network

Table 2.14 Link flow and capacity allocation of a 20-node network

Link	Capacity	Traffic flow (calls/min)	
	(leased line)	Leased line	The third party
(1,2)	1DS1	3.757	0.009
(1,3)	2 DS2	46.443	0.117
(1,4)	1 DS1 + 1 DS2	30.813	1.056
(2,3)	4 DS1 + 3 DS2	97.14	0.138
(2,17)	—	—	0.194
(3,4)	2 DS1	8.4	0.004
(3,5)	13 DS1 + 3 DS2	153.962	0.092
(3,6)	2 DS2	49.172	0.668
(3,7)	—	—	0.050
(3,8)	—	—	0.117
(4,5)	5 DS1 + 3 DS2	105.203	0.299
(4,15)	—	—	0.121
(5,6)	2 DS1 + 1 DS2	34.194	0.124
(5,7)	17 DS1 + 1 DS3	284.507	0.112
(6,7)	11 DS1 + 3 DS2	140.281	0.63
(6,17)	—	—	0.127
(7,8)	2 DS1 + 3 DS2	80.637	0.014
(7,10)	5 DS1 + 1 DS3	215.806	2.036
(7,11)	14 DS1 + 3 DS2	162.172	0.185
(8,9)	2 DS2	44.165	0.019

Table 2.14 Link flow and capacity allocation of a 20-node network (cont'd)

Link	Capacity (leased line)	Traffic flow (calls/min)	
		Leased line	The third party
(8,10)	3 DS2	69.374	0.032
(9,10)	3 DS0+2 DS1+1 DS2	37.394	0.941
(9,14)	—	—	0.049
(10,11)	2 DS1 + 2 DS2	59.926	0.219
(10,12)	3 DS2	68.971	0.024
(10,14)	2 DS2	47.301	0.211
(10,16)	12 DS1 + 3 DS2	149.616	0.216
(11,12)	1 DS1 + 3 DS2	76.329	0.050
(11,13)	4 DS1 + 1 DS3	207.635	1.337
(12,14)	1DS2	17.800	0.0
(12,16)	1 DS1 + 2 DS2	52.581	0.112
(13,15)	1 DS3	172.117	0.056
(14,16)	2 DS1 + 1 DS2	35.6	0.361
(15,17)	1 DS1 + 3 DS2	77.377	0.094
(15,18)	2 DS2	49.397	0.761
(15,20)	2 DS1 + 2 DS2	62.629	0.947
(16,18)	1 DS3	178.352	0.554
(17,20)	—	—	0.460
(18,19)	7 DS1 + 3 DS2	114.952	0.072
(19,20)	1 DS1 + 1 DS2	30.492	0.835

Table 2.15 Routing fractions of a 20-node network

Source	Destination	Link	Fraction
1	2	(1,2)	1.0000
	3	(1,3)	1.0000
	4	(1,4)	1 0000
	5	(1,3)	0 4113
		(1,4)	0 5887
	6	(1,3)	0 7364
		(1,4)	0 2636
	7	(1,3)	0 1592
		(1,4)	0.8408
	8	(1,3)	0.8694
		(1,4)	0.1306
	9	(1,3)	0 1413
		(1,4)	0.8587
	10	(1,4)	1 0000
	11	(1,4)	1 0000
	12	(1,3)	0 2951
		(1,4)	0 7049
	13	(1,3)	0 3315
		(1,4)	0 6685
	14	(1,3)	0.1010
		(1,4)	0 8990
	15	(1,4)	1.0000
	16	(1,3)	0.3629
		(1,4)	0 6371
	17	(1,2)	0 0010
		(1,4)	0 9990
	18	(1,4)	1 0000
	19	(1,4)	1.0000
	20	(1,4)	1 0000
2	1	(2,1)	1.0000
	3	(2,3)	1.0000
	4	(2,3)	1 0000
	5	(2,3)	1 0000
	6	(1,2)	0.3578
		(2,3)	0.6422
	7	(2,3)	1 0000
	8	(2,3)	1.0000
	9	(2,3)	1.0000
	10	(1,2)	0.2516
		(2,3)	0.7484
	11	(1,2)	0.1108
		(2,3)	0.8892
	12	(2,3)	1.0000
	13	(2,3)	1.0000
	14	(2,3)	1.0000
	15	(2,3)	1.0000
	16	(2,3)	1.0000
	17	(2,3)	0.9979
		(2,17)	0.0021
	18	(2,3)	1.0000
	19	(2,3)	1.0000
	20	(2,3)	1.0000

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
3	1	(1,3)	1 0000
	2	(2,3)	1 0000
	4	(3,4)	1 0000
	5	(3,5)	1.0000
	6	(3,6)	1.0000
	7	(3,5)	0 8502
		(3,6)	0 1486
		(3,7)	0 0012
	8	(3,5)	0 4195
		(3,6)	0 5700
		(3,8)	0.0105
	9	(3,5)	0.5947
		(3,6)	0 4053
	10	(3,5)	1.0000
	11	(3,5)	1 0000
	12	(3,5)	1 0000
	13	(3,5)	1 0000
	14	(3,6)	1 0000
	15	(3,5)	1 0000
	16	(3,6)	1 0000
	17	(3,5)	0 7351
		(3,6)	0 2649
	18	(3,6)	1.0000
	19	(3,6)	1 0000
	20	(3,6)	1 0000

Source	Destination	Link	Fraction
4	1	(1,4)	1 0000
	2	(3,4)	1 0000
	3	(3,4)	1 0000
	5	(4,5)	1 0000
	6	(4,5)	1 0000
	7	(4,5)	1 0000
	8	(4,5)	1 0000
	9	(4,5)	1 0000
	10	(4,5)	1 0000
	11	(4,5)	1 0000
	12	(4,5)	1 0000
	13	(4,5)	1 0000
	14	(4,5)	1 0000
	15	(4,5)	0 9939
		(4,15)	0 0061
	16	(4,5)	1 0000
	17	(4,5)	0 9900
		(4,15)	0 0100
	18	(4,5)	0 9977
		(4,15)	0 0023
	19	(4,5)	1 0000
	20	(4,5)	1.0000

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
5	1	(5,3)	1 0000
	2	(5,3)	1.0000
	3	(5,3)	1 0000
	4	(5,4)	1 0000
	6	(5,6)	1 0000
	7	(5,7)	1 0000
	8	(5,6)	0.3081
		(5,7)	0.6919
	9	(5,7)	1 0000
	10	(5,6)	0 2915
		(5,7)	0.7085
	11	(5,7)	1.0000
	12	(5,7)	1 0000
	13	(5,7)	1 0000
	14	(5,7)	1.0000
	15	(5,7)	1 0000
	16	(5,7)	1 0000
	17	(5,7)	1.0000
	18	(5,7)	1.0000
	19	(5,7)	1.0000
	20	(5,7)	1 0000
6	1	(6,3)	0.8190
		(6,5)	0.1810
	2	(6,3)	0.7261
		(6,5)	0.2739
	3	(6,3)	1.0000
	4	(6,5)	1.0000
	5	(6,5)	1.0000
	7	(6,7)	1.0000
	8	(6,7)	1.0000
	9	(6,7)	1.0000
	10	(6,7)	1.0000
	11	(6,7)	1.0000
	12	(6,7)	1.0000
	13	(6,7)	1.0000
	14	(6,7)	1.0000
	15	(6,7)	0.9884
		(6,17)	0.0116
	16	(6,7)	1 0000
	17	(6,7)	0.9880
		(6,17)	0.0120
	18	(6,7)	1.0000
	19	(6,7)	1.0000
	20	(6,7)	0.9968
		(6,17)	0.0032

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction	Source	Destination	Link	Fraction
	1	(7,3)	0.0013		1	(8,3)	0 0109
		(7,5)	0.6012			(8,7)	0 9891
		(7,6)	0 3975		2	(8,7)	1 0000
	2	(7,5)	1.0000		3	(8,3)	0 0008
	3	(7,5)	1 0000			(8,7)	0 9992
	4	(7,5)	0.5024		4	(8,7)	1 0000
		(7,6)	0 4976		5	(8,7)	1 0000
	5	(7,5)	1.0000		6	(8,7)	1 0000
	6	(7,6)	1.0000		7	(8,7)	1 0000
	8	(7,8)	1 0000		9	(8,9)	1 0000
	9	(7,8)	1 0000	8	10	(8,10)	1 0000
7	10	(7,10)	1.0000		11	(8,10)	1 0000
	11	(7,11)	1 0000		12	(8,10)	1 0000
	12	(7,10)	0.6722		13	(8,10)	1 0000
		(7,11)	0 3278		14	(8,9)	0 0015
	13	(7,10)	1 0000			(8,10)	0 9985
	14	(7,10)	1 0000		15	(8,10)	1 0000
	15	(7,11)	1 0000		16	(8,9)	0 0006
	16	(7,10)	0 5167			(8,10)	0 9994
		(7,11)	0 4833		17	(8,10)	1 0000
	17	(7,11)	1.0000		18	(8,10)	1 0000
	18	(7,10)	1 0000		19	(8,10)	1 0000
	19	(7,10)	1.0000		20	(8,10)	1 0000
	20	(7,11)	1 0000				

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction	Source	Destination	Link	Fraction
9	1	(9,8)	1.0000	10	1	(10,7)	1.0000
	2	(9,8)	1.0000		2	(10,7)	1.0000
	3	(9,8)	1.0000		3	(10,8)	1.0000
	4	(9,8)	1.0000		4	(10,7)	1.0000
	5	(9,8)	1.0000		5	(10,7)	1.0000
	6	(9,8)	1.0000		6	(10,7)	1.0000
	7	(9,8)	1.0000		7	(10,7)	1.0000
	8	(9,8)	1.0000		8	(10,8)	1.0000
	10	(9,10)	1.0000		9	(10,9)	1.0000
	11	(9,10)	1.0000		11	(10,11)	1.0000
	12	(9,10)	1.0000		12	(10,12)	1.0000
	13	(9,10)	1.0000		13	(10,11)	1.0000
	14	(9,10)	0.9969		14	(10,14)	1.0000
		(9,14)	0.0031		15	(10,11)	1.0000
	15	(9,10)	1.0000		16	(10,16)	1.0000
	16	(9,10)	0.9999		17	(10,11)	1.0000
		(9,14)	0.0001		18	(10,16)	1.0000
	17	(9,10)	1.0000		19	(10,16)	1.0000
	18	(9,10)	1.0000		20	(10,11)	0.6114
	19	(9,10)	1.0000			(10,16)	0.3886
	20	(9,10)	1.0000				

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction	Source	Destination	Link	Fraction
11	1	(11,7)	1.0000	12	1	(12,10)	1.0000
	2	(11,7)	1.0000		2	(12,10)	1.0000
	3	(11,7)	0.9922		3	(12,10)	1.0000
		(11,10)	0.0078		4	(12,10)	1.0000
	4	(11,7)	1.0000		5	(12,10)	1.0000
	5	(11,7)	0.8592		6	(12,10)	1.0000
		(11,10)	0.1408		7	(12,10)	1.0000
	6	(11,7)	1.0000		8	(12,10)	1.0000
	7	(11,7)	1.0000		9	(12,10)	1.0000
	8	(11,10)	1.0000		10	(12,10)	1.0000
	9	(11,10)	1.0000		11	(12,11)	1.0000
	10	(11,10)	1.0000		13	(12,11)	1.0000
	12	(11,12)	1.0000		14	(12,14)	1.0000
	13	(11,13)	1.0000		15	(12,11)	1.0000
	14	(11,12)	1.0000		16	(12,16)	1.0000
	15	(11,13)	1.0000		17	(12,16)	1.0000
	16	(11,12)	1.0000		18	(12,16)	1.0000
	17	(11,13)	1.0000		19	(12,16)	1.0000
	18	(11,12)	0.7409		20	(12,16)	1.0000
		(11,13)	0.2591				
	19	(11,12)	1.0000				
	20	(11,13)	1.0000				

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
13	1	(13,11)	1.0000
	2	(13,11)	1 0000
	3	(13,11)	1 0000
	4	(13,11)	1.0000
	5	(13,11)	1 0000
	6	(13,11)	1 0000
	7	(13,11)	1 0000
	8	(13,11)	1.0000
	9	(13,11)	1 0000
	10	(13,11)	1 0000
	11	(13,11)	1.0000
	12	(13,11)	1 0000
	14	(13,11)	1.0000
	15	(13,15)	1 0000
	16	(13,15)	1 0000
	17	(13,15)	1.0000
	18	(13,15)	1 0000
	19	(13,15)	1.0000
	20	(13,15)	1.0000
14	1	(14,9)	0 0014
		(14,10)	0.9986
	2	(14,10)	1.0000
	3	(14,9)	0.0008
		(14,10)	0.9992
	4	(14,10)	1.0000
	5	(14,10)	1.0000
	6	(14,10)	1.0000
	7	(14,10)	1.0000
	8	(14,9)	0.0109
		(14,10)	0.9891
	9	(14,9)	0.0112
		(14,10)	0.9888
	10	(14,10)	1 0000
	11	(14,12)	1 0000
	12	(14,12)	1.0000
	13	(14,12)	1.0000
	15	(14,16)	1.0000
	16	(14,16)	1.0000
	17	(14,16)	1.0000
	18	(14,16)	1.0000
	19	(14,16)	1.0000
	20	(14,16)	1.0000

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
15	1	(15,4)	0 0103
		(15,13)	0.9897
	2	(15,4)	0 0005
		(15,13)	0 9995
	3	(15,4)	0 0002
		(15,13)	0 9998
	4	(15,4)	0 0018
		(15,13)	0 9982
	5	(15,13)	1 0000
	6	(15,13)	1.0000
	7	(15,13)	1.0000
	8	(15,13)	1.0000
	9	(15,13)	1 0000
	10	(15,13)	1 0000
	11	(15,13)	1.0000
	12	(15,13)	1.0000
	13	(15,13)	1.0000
	14	(15,18)	1 0000
	16	(15,18)	1.0000
	17	(15,17)	1 0000
	18	(15,18)	1 0000
	19	(15,18)	1 0000
	20	(15,20)	1 0000

Source	Destination	Link	Fraction
16	1	(16,10)	0 8427
		(16,12)	0 1573
	2	(16,10)	1 0000
	3	(16,10)	0 6329
		(16,12)	0 3671
	4	(16,10)	1 0000
	5	(16,10)	1 0000
	6	(16,10)	1 0000
	7	(16,10)	1 0000
	8	(16,10)	1 0000
	9	(16,10)	0 9114
		(16,14)	0 0886
	10	(16,10)	1 0000
	11	(16,12)	1 0000
	12	(16,12)	1 0000
	13	(16,12)	1.0000
	14	(16,14)	1 0000
	15	(16,18)	1 0000
	17	(16,18)	1 0000
	18	(16,18)	1 0000
	19	(16,18)	1 0000
	20	(16,18)	1 0000

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
17	1	(17,2)	0 1002
		(17,15)	0 8998
	2	(17,2)	0 0003
		(17,15)	0 9997
	3	(17,15)	1 0000
	4	(17,15)	1 0000
	5	(17,15)	1 0000
	6	(17,6)	0 0111
		(17,15)	0 9889
	7	(17,15)	1 0000
	8	(17,15)	1.0000
	9	(17,15)	1.0000
	10	(17,15)	1 0000
	11	(17,15)	1 0000
	12	(17,15)	1 0000
	13	(17,15)	1 0000
	14	(17,15)	1 0000
	15	(17,15)	1 0000
	16	(17,15)	1 0000
	18	(17,15)	1.0000
	19	(17,15)	0 9062
		(17,20)	0 0938
	20	(17,15)	0 9485
		(17,20)	0 0315

Source	Destination	Link	Fraction
18	1	(18,16)	1 0000
	2	(18,16)	1 0000
	3	(18,15)	0.1574
		(18,16)	0.8426
	4	(18,15)	0 1629
		(18,16)	0 8371
	5	(18,16)	1 0000
	6	(18,16)	1 0000
	7	(18,16)	1 0000
	8	(18,16)	1 0000
	9	(18,16)	1.0000
	10	(18,16)	1 0000
	11	(18,16)	1 0000
	12	(18,16)	1 0000
	13	(18,15)	1 0000
	14	(18,16)	1 0000
	15	(18,15)	1.0000
	16	(18,16)	1 0000
	17	(18,15)	1 0000
	19	(18,19)	1 0000
	20	(18,19)	1.0000

Table 2.15 Routing fractions of a 20-node network (cont'd)

Source	Destination	Link	Fraction
19	1	(19,18)	1 0000
	2	(19,18)	1 0000
	3	(19,18)	1 0000
	4	(19,18)	1 0000
	5	(19,18)	1 0000
	6	(19,18)	1 0000
	7	(19,18)	1 0000
	8	(19,18)	1 0000
	9	(19,18)	1 0000
	10	(19,18)	1 0000
	11	(19,18)	1 0000
	12	(19,18)	1 0000
	13	(19,18)	1 0000
	14	(19,18)	1 0000
	15	(19,18)	1.0000
	16	(19,18)	1 0000
	17	(19,18)	0 3058
		(19,20)	0.6942
	18	(19,18)	1 0000
	20	(19,20)	1 0000

Source	Destination	Link	Fraction
20	1	(20,15)	0 9876
		(20,17)	0 0124
	2	(20,15)	0 9855
		(20,17)	0.0145
	3	(20,15)	1 0000
	4	(20,15)	1 0000
	5	(20,15)	1 0000
	6	(20,15)	1 0000
	7	(20,15)	1 0000
	8	(20,15)	0 6913
		(20,19)	0 3087
	9	(20,19)	1 0000
	10	(20,19)	1 0000
	11	(20,15)	1 0000
	12	(20,15)	1 0000
	13	(20,15)	1 0000
	14	(20,19)	1 0000
	15	(20,15)	1 0000
	16	(20,19)	1 0000
	17	(20,15)	0 9933
		(20,17)	0 0067
	18	(20,19)	1 0000
	19	(20,19)	1 0000

Chapter 3

OPTIMAL ROUTING AND DIMENSIONING FOR MULTIPLE-SERVICE CIRCUIT-SWITCHED NETWORKS

We restrict our attention to circuit-switched networks that support a variety of traffic classes with different traffic characteristics—bandwidth requirement, call arrival rate and call holding time. The composition of traffic transmitted through links is likely to consist of mixtures of voice, facsimile, video, etc. The different traffic types will place different requirements on the system design.

The problem of routing calls on a shared transport network has been considered as the problem of controlling a stochastic network with finite capacity that can process calls of different types [22][23]. The network can be represented by a finite state Markov chain. Since each call that is carried in the network generates a known revenue, a minimization of the lost revenue is chosen as a network performance measure. Thus, application of Markov decision theory, with the objective of optimizing the average cost, is appropriate for this problem. A major difficulty in applying Markov decision theory to solve multirate routing problem is the computational complexity. Therefore, the purpose in the literature [22][23] has been to find computationally tractable methods for determining nearly optimal policies.

Our attempt is to extend the formulation of optimization problem for single-service networks presented in last chapter to multiple-service networks. For the sake of simplicity, the following chapter considers the case of having two traffic classes—voice and video. The arguments given in the chapter are also valid for cases with more than two traffic classes.

It is assumed that overflow traffic of either type on a given link can be carried by a third party and that, again, call routes may belong partly to the network and to the third party. In this heterogeneous environment, analytical models are employed to determine link blocking. Similar to the case of voice-carrying networks in Chapter 2, a multivariate composite cost function is constructed to allocate a minimum-cost capacity over each link.

3.1 THE FRAMEWORK

3.1.1 Elements of the Problem

The network under consideration has N nodes. Initially, the network is assumed to be fully connected. There are $N(N-1)/2$ bidirectional links. The case of networks with one-way links is not included because we expect that realistic networks will operate with two-way links.

Traffic is divided into two classes of service, class 1 corresponding to voice traffic and class 2 corresponding to video traffic. For each class of service $k(k = 1,2)$ we make the following assumptions:

(i) For an O-D pair (i,j) calls arrive according to a stationary Poisson process with a mean arrival rate $\lambda_i^k(j)$.

(ii) Call holding times are independent and exponentially distributed with the mean $1/\mu_k$.

(iii) Each call has a bit rate or capacity requirement. A voice call needs a DS0 rate (64 kbps, 1 circuit), and a video call needs a DS1 rate (1.544 Mbps, 24 circuits).

Any call offered to the network is either accepted or carried by a third party. To measure the amount of traffic carried by the third party, blocking of each type of call on links must be determined. Consider a link having capacity in terms of K circuits to which two types of traffic are offered, the problem now is to find approximations for the blocking probabilities B^1 and B^2 for the two traffic streams. Earlier studies of such kinds of problems are reported in [43][44][45][46]. Let $P(n_1, ..., n_k)$ be the probability distribution of the number of calls present from each class, then P has the product form:

$$P(n_1, ..., n_k) = \Pi_{i=1}^{k} \frac{a_i^{n_i}}{n_i!} \cdot P(0,0) \tag{3.1}$$

where $a_i = \lambda_i/\mu_i$ and $P(0,0)$ is given by a normalizing condition. The call holding time need not be exponentially distributed for this formula to hold. Kaufman [47] derived a simple recurrence relation that is satisfied by the distribution of the total number of circuits occupied. Let this distribution be $q(j)$, j=total number of circuits occupied, then the recursion is given by

$$\sum_{i=1}^{k} a_i b_i q(j - b_i) = jq(j)\,, \qquad j = 0, 1, ..., K \tag{3.2}$$

where $q(x) = 0$ for $x < 0$ and

$$\sum_{j=0}^{K} q(j) = 1 \tag{3.3}$$

The blocking probabilities for each type of call are given by

$$B^i = \sum_{i=0}^{b_i-1} q(K-i)\,, \qquad i = 1,..,k \tag{3.4}$$

For our case, $k = 2$, $b_1 = 1$, $b_2 = 24$.

3.1.2 Formulation of the Model

Let $r^1_{il}(j)$ and $r^2_{il}(j)$ be the fraction of all voice, video calls at node i and destined for node j that is routed over link (i,l) respectively. Flow conservation for each type is expressed with a separated traffic equation

$$\begin{aligned} \Lambda^1_i(j) &= \lambda^1_i(j) + \textstyle\sum_l \Lambda^1_l(j) r^1_{li}(j) \\ \Lambda^2_i(j) &= \lambda^2_i(j) + \textstyle\sum_l \Lambda^2_l(j) r^2_{li}(j) \end{aligned} \tag{3.5}$$

where the routing variables satisfy

$$0 \leq r^1_{il}(j), \quad r^2_{il}(j) \leq 1 \quad \text{and}$$

$$\textstyle\sum_l r^1_{il}(j) = \sum_l r^2_{il}(j) = 1, \quad i,j \in \{1,...,N\}$$

The link loads consist of two components ρ^1_{ij} and ρ^2_{ij}, given by

$$\begin{aligned} \rho^1_{ij} &= [\textstyle\sum_{l,l\neq i} \Lambda^1_i(l) r^1_{ij}(l) + \sum_{l,l\neq j} \Lambda^1_j(l) r^1_{ji}(l)]/\mu_1 \\ \rho^2_{ij} &= [\textstyle\sum_{l,l\neq i} \Lambda^2_i(l) r^2_{ij}(l) + \sum_{l,l\neq j} \Lambda^2_j(l) r^2_{ji}(l)]/\mu_2 \end{aligned} \tag{3.6}$$

Both load components are used to compute a cost for a link. The cost of link (i,j) accounts for the fixed leasing fee as well as the cost incurred by having all overflow traffic carried by a third party, it is in the form

$$C_{ij}(\rho^1_{ij}, \rho^2_{ij}, d_{ij}, c_{ij}, o_{ij}) = C_1(c_{ij}, d_{ij}) + \alpha^1_{ij} o^1_{ij} + \alpha^2_{ij} o^2_{ij} \tag{3.7}$$

where $C_1(c_{ij}, d_{ij})$ is given by (2.3) , and the overflow components are given by

$$o_{ij}^1 = 189\rho_{ij}^1 B^1(\rho_{ij}^1, \rho_{ij}^2, c_{ij})$$
$$o_{ij}^2 = 189\rho_{ij}^2 B^2(\rho_{ij}^1, \rho_{ij}^2, c_{ij}) \quad (3.8)$$

The parameters α_{ij}^1 and α_{ij}^2, which are given in Table 2.5 and Table 3.1 respectively, are the cost in \$/erlang for carrying respective types of overflow traffic between nodes i and j. $B^1(\cdot)$ and $B^2(\cdot)$ measure the blocking probabilities on link (i, j) by Kaufman's blocking model [47].

Table 3.1 Rate of carrying the overflow video traffic

Distance (miles)	Rates (\$/erlang)
1 – 60	195.0
61 – 400	230.00
401 – 1000	295.00
1001 – 2000	316.0
over 2000	343.0

The value of c_{ij}^* is again chosen to minimize (3.7) over all possible capacity assignments at each operating point $(\rho_{ij}^1, \rho_{ij}^2)$. The cost function can be written

$$C_{ij}(\rho_{ij}^1, \rho_{ij}^2, d_{ij}, c_{ij}^*, o_{ij}) = \begin{cases} \underbrace{\min}_{c_{ij} \in \{DS0, \ldots DS4\}} \{C_{ij}(\rho_{ij}^1, \rho_{ij}^2, d_{ij}, c_{ij}, o_{ij})\} & i \neq j \\ 0 & i = j \end{cases} \quad (3.9)$$

where ρ^1_{ij}, $\rho^2_{ij} \in [0, \infty)$. Hence, the cost function assigns a least-cost solution independently for each link. The cost function of a link distance $d_{ij} = 75$ miles is shown in Fig. 3.1.

The total cost for the network is

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}(\rho^1_{ij}, \rho^2_{ij}, d_{ij}, c^*_{ij}, o_{ij}) \tag{3.10}$$

Our algorithm minimizes the total cost of the network over the routing variables $r^1_{il}(j)$ and $r^2_{il}(j)$. The network is dimensioned by assigning a unique c^*_{ij} that satisfies the resulting link loads for each link.

3.1.3 The Algorithm

The algorithm for the multirate case proceeds exactly as described in section 2.3, but maintains analogous sets of variables for each type of traffic.

Let

$SP^k_l(j)$: the shortest path for type k traffic from node l to node j

D^k_{ij} : the length of link (i, j) for type k traffic

$D^k_i(j)$: the length of the shortest path from node l to j for type k traffic

$\Delta^k_{il}(j)$: the amount of reduction of $r^k_{il}(j)$

where $k = 1, 2$. The length for each link

$$\begin{aligned} D^1_{ij} &= \frac{\partial C^1_{ij}(\rho^1_{ij}, \rho^2_{ij}, d_{ij}, c^*_{ij}, o^1_{ij})}{\partial \rho^1_{ij}} \\ D^2_{ij} &= \frac{\partial C^2_{ij}(\rho^1_{ij}, \rho^2_{ij}, d_{ij}, c^*_{ij}, o^2_{ij})}{\partial \rho^2_{ij}} \end{aligned} \tag{3.11}$$

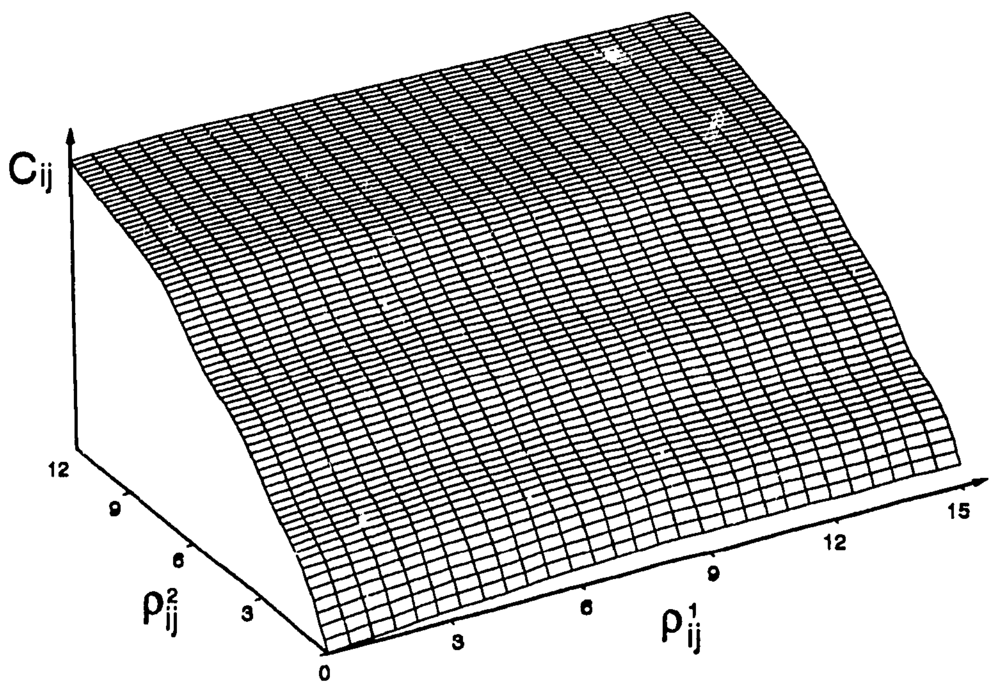


Fig. 3.1 Compound link cost function

The algorithm is initialized with

$$r^1_{il}(j) = r^2_{il}(j) = \begin{cases} 1 & if \ \ l = j \\ 0 & otherwise \end{cases} \tag{3.12}$$

On each iteration, the algorithm updates the current routing variables according to

$$r^k_{il}(j) = \begin{cases} r^k_{il}(j) - \Delta^k_{il}(j) & if \ \ (i,l) \notin SP^k_i(j) \\ r^k_{il}(j) + \sum_{(i,l) \notin SP^k_i(j)} \Delta^k_{il}(j) & if \ \ (i,l) \in SP^k_i(j) \end{cases} \tag{3.13}$$

for all $l \in \{1, ..., N\}$, $k = 1, 2$ and

$$\Delta^k_{il}(j) = \min\left\{ r^k_{il}(j), \eta[1 - \frac{D^k_i(j)}{D^k_{il} + D^k_l(j)}] \right\} \tag{3.14}$$

η is a scale parameter, it is chosen by the same form of line search described in section 2.4.

The algorithm terminates after the kth iteration if

$$\begin{aligned} [r^1_{il}(j)]^{k+1} &= [r^1_{il}(j)]^k \\ &\& \\ [r^2_{il}(j)]^{k+1} &= [r^2_{il}(j)]^k \quad \forall i, j, l \in \{1, .., N\} \end{aligned} \tag{3.15}$$

The procedure of the algorithm is expressed below

```
Initialization;
WHILE (termination criteria are not met) {
    FOR j = 1 to N {
        FOR i = 1 to N {
            /* update all node and link flows by (3.5), (3.6) */
            /* calculate D^1_ij, D^2_ij by (3.11) and
               determine SP^1_i(j), SP^2_i(j) for all i ∈ {1,..., N} */
            /* update routing variables */
            /* test termination conditions */
        }
    }
}
```

3.2 NUMERICAL RESULTS

To present some examples, we consider a network that has $N = 6$ nodes. Between each O-D pair there are two types of traffic service: voice and video. The call holding time for voice traftic $1/\mu_1 = 3.5$ mins, for video traffic $1/\mu_2 = 30$ mins. Table 3.2 and Table 3.3 give the exogenous arrivals of voice calls and video calls. The distance matrix is specified in Table 3.4.

We use the network cost as a measure of network performance. Our result (Fig. 3.2 and Table 3.3) shows that a marked decrease from the initial network cost is obtained. We can see clearly in Fig. 3.3 that paths per O-D pair for voice or video traffic are composed of links belonging to the network and links belonging

Table 3.2 Original voice traffic matrix (calls/min)

$\lambda_i^1(j)$	1	2	3	4	5	6
1	—	8.0	10.4	1.4	0.8	3.6
2	7.5	—	4.8	10.5	24.3	2.1
3	10.0	5.2	—	3.1	10.5	1.0
4	1.1	9.8	2.6	—	2.2	1.8
5	1.2	21.8	9.2	2.8	—	4.3
6	3.2	1.75	1.45	2.3	5.0	—

Table 3.3 Original video traffic matrix (calls/day)

$\lambda_i^1(j)$	1	2	3	4	5	6
1	—	5	5	2	3	1
2	3	—	4	3	5	1
3	2	3	—	1	2	1
4	3	5	3	—	4	4
5	2	6	2	3	—	3
6	5	5	4	2	2	—

Table 3.4 Distance matrix of a 6-node network (miles)

d_{ij}	1	2	3	4	5	6
1	—	1132.0	2807.5	2872.0	2882.0	3000.0
2		—	1312.0	1372.0	1386.0	1505.0
3			—	57.0	249.0	337.0
4				—	228.0	345.0
5					—	118.0
6						—

to the third party. The traffic on a link will be carried by the third party only, when it is very light. In another example, external arrivals of voice and video calls are identical at each node, say 3 calls/min for voice traffic and 2 calls/day for video traffic. Exhibited in Fig. 3.4 and Table 3.4, Table 3.5, paths of each O-D pair for voice and video traffic favour links with shorter physical distance.

For both examples the algorithm converged to the solutions within 2000 iterations. Through many experiments, we saw that for small networks (nodes of a network ≤ 7) the algorithm may converge in a shorter period, and for the large networks the algorithm may require very long CPU time, all runs for a 20-nodes network (locations of nodes shown in Fig. 2.13) with different external traffic matrix converged to the steady state in four hours. The strength of our approach is that it provides a simple and easily to implemented solution to the routing and dimensioning problem for multiple-service circuit-switched networks. It gives a framework for integrated networks planning.

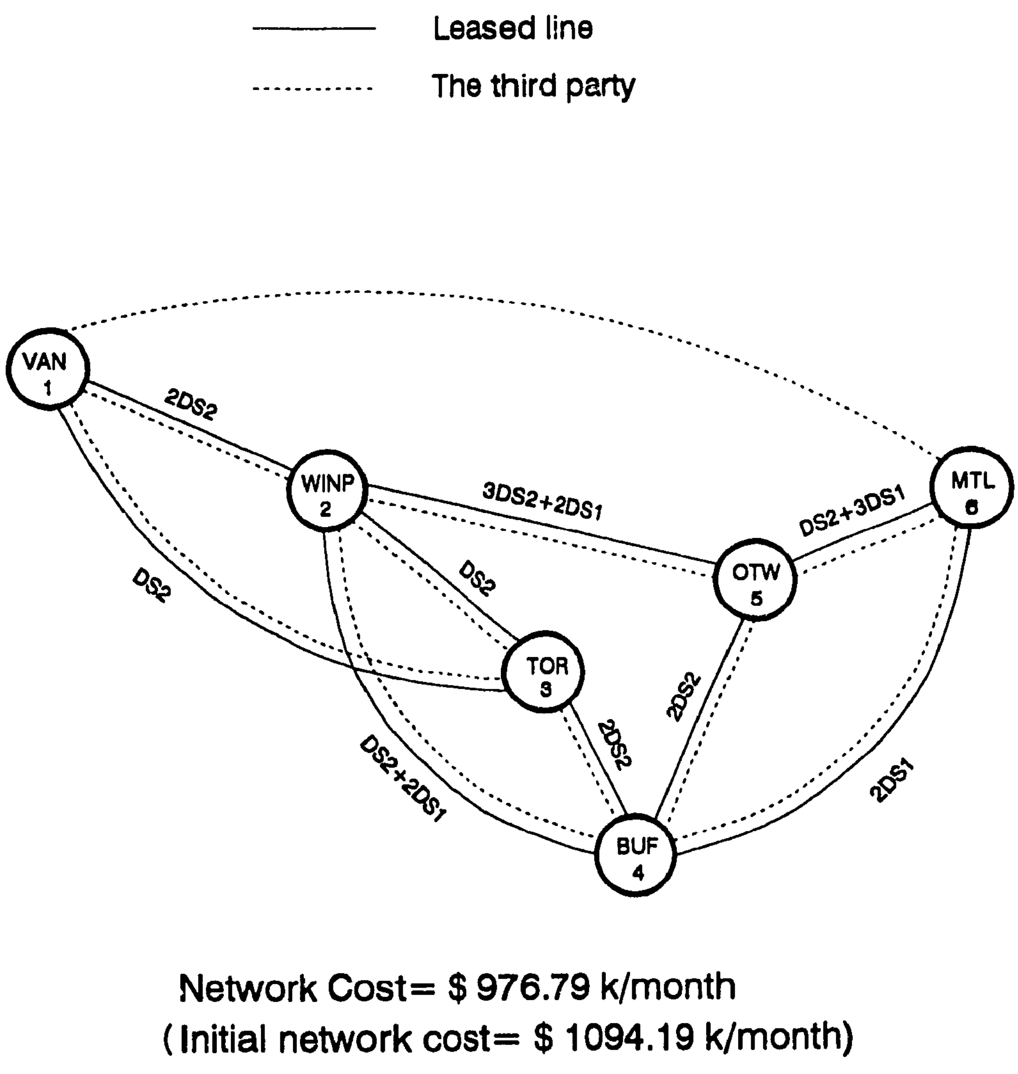


Fig. 3.2 The final physical topology of the 6-node network (1)

Table 3.5 Link flow of a 6-node network(1)

Link	Leased line		Third party	
	Voice (calls/min)	Video (calls/day)	Voice (calls/min)	Video (calls/day)
(1,2)	31.1149	16.6486	0.0521	1.3514
(1,3)	15.9793	5.2366	0.0537	1.7634
(1,6)	—	—	—	1.0
(2,3)	15.4889	5.3662	0.0478	1.6338
(2,4)	21.5918	11.7472	0.0384	1.2579
(2,5)	58.7375	21.7599	0 0125	0.2401
(3,4)	27.841	12.7398	0.009	0.2602
(4,5)	28.8424	15.3688	0.0208	0.6312
(4,6)	3.8837	4.488	0.0031	1 512
(5,6)	22.5965	15.3075	0.0166	0 6925

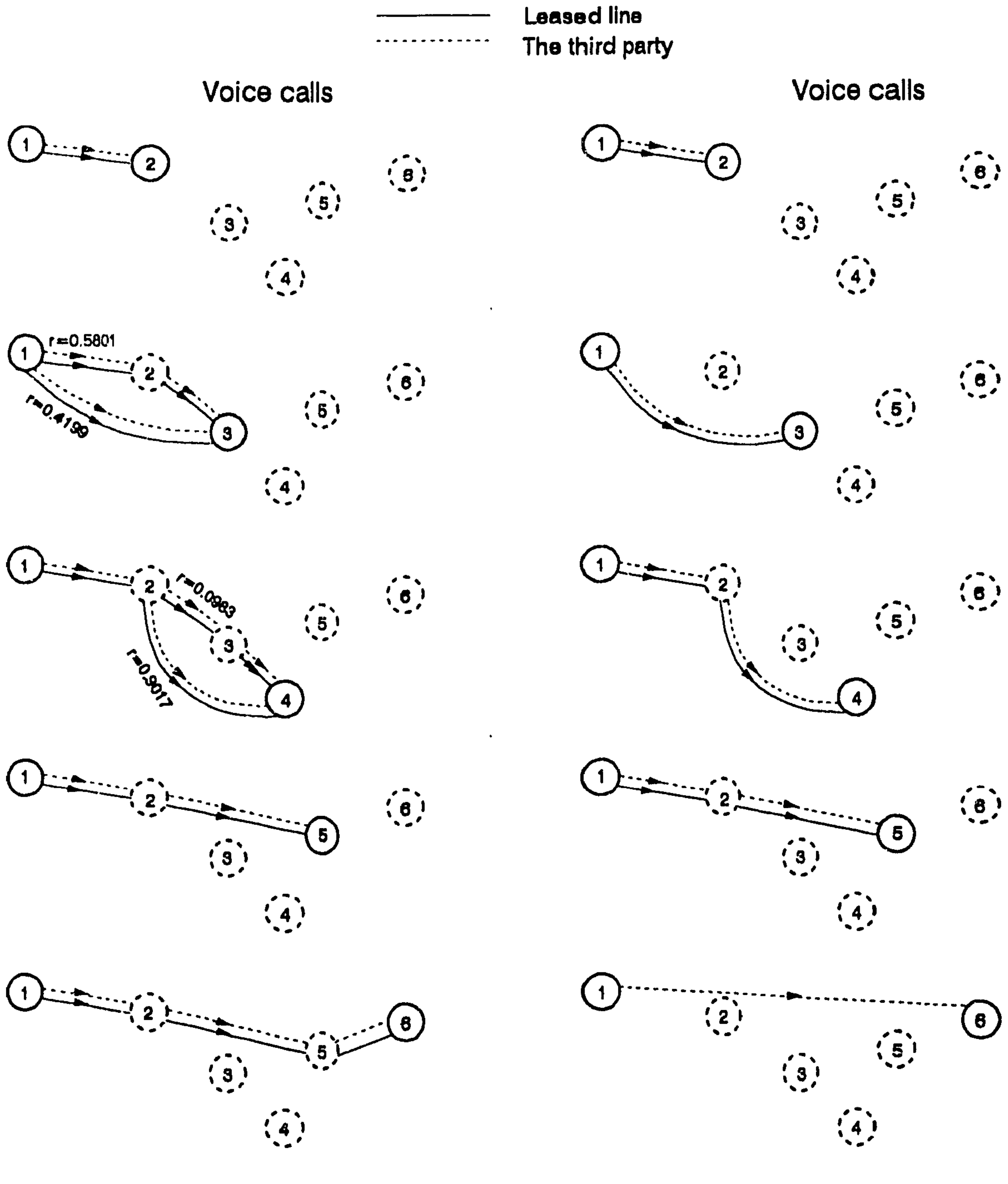


Fig. 3.3 The routes of the traffic for each O-D pair

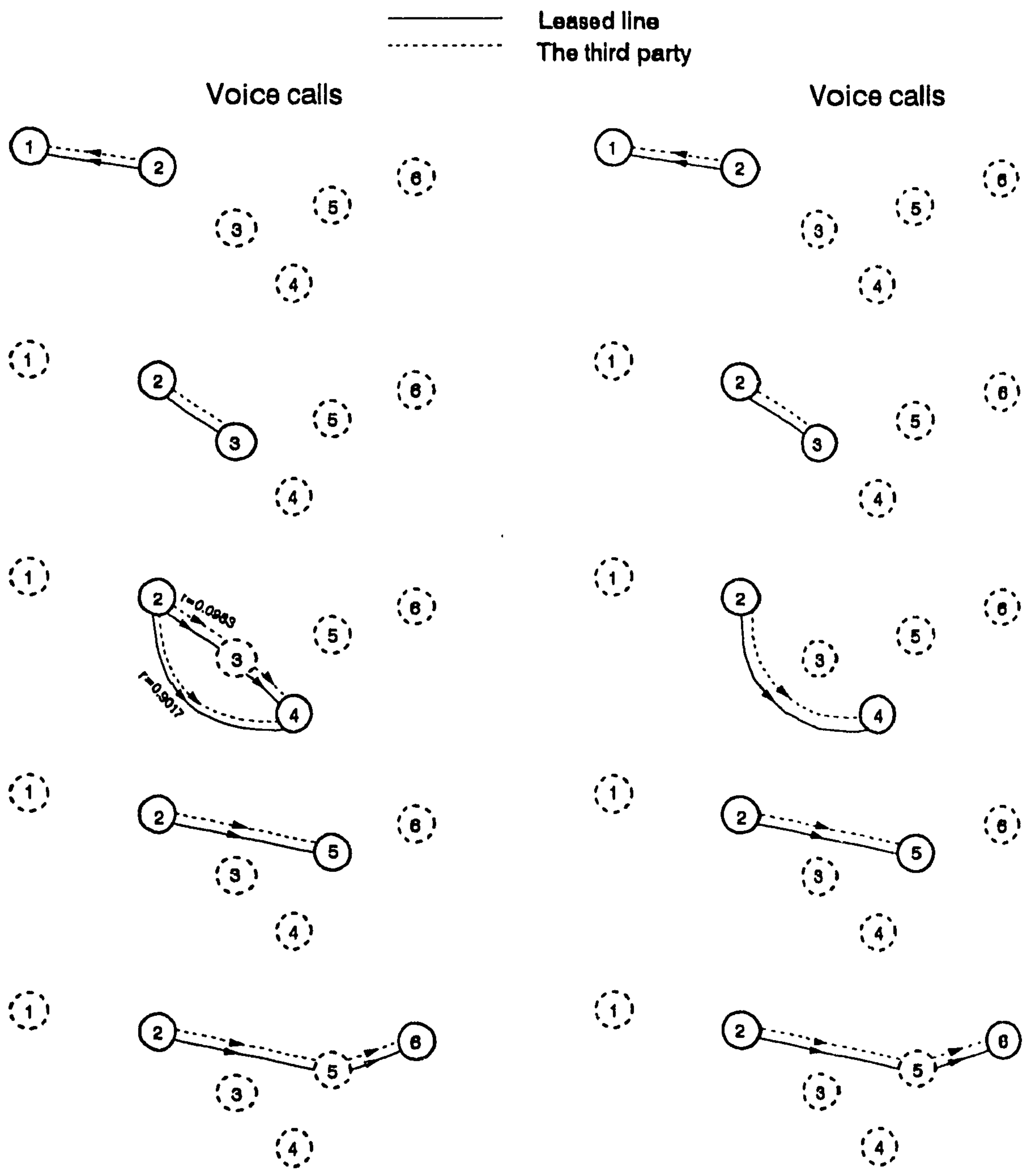


Fig. 3.3 The routes of the traffic for each O-D pair (cont'd)

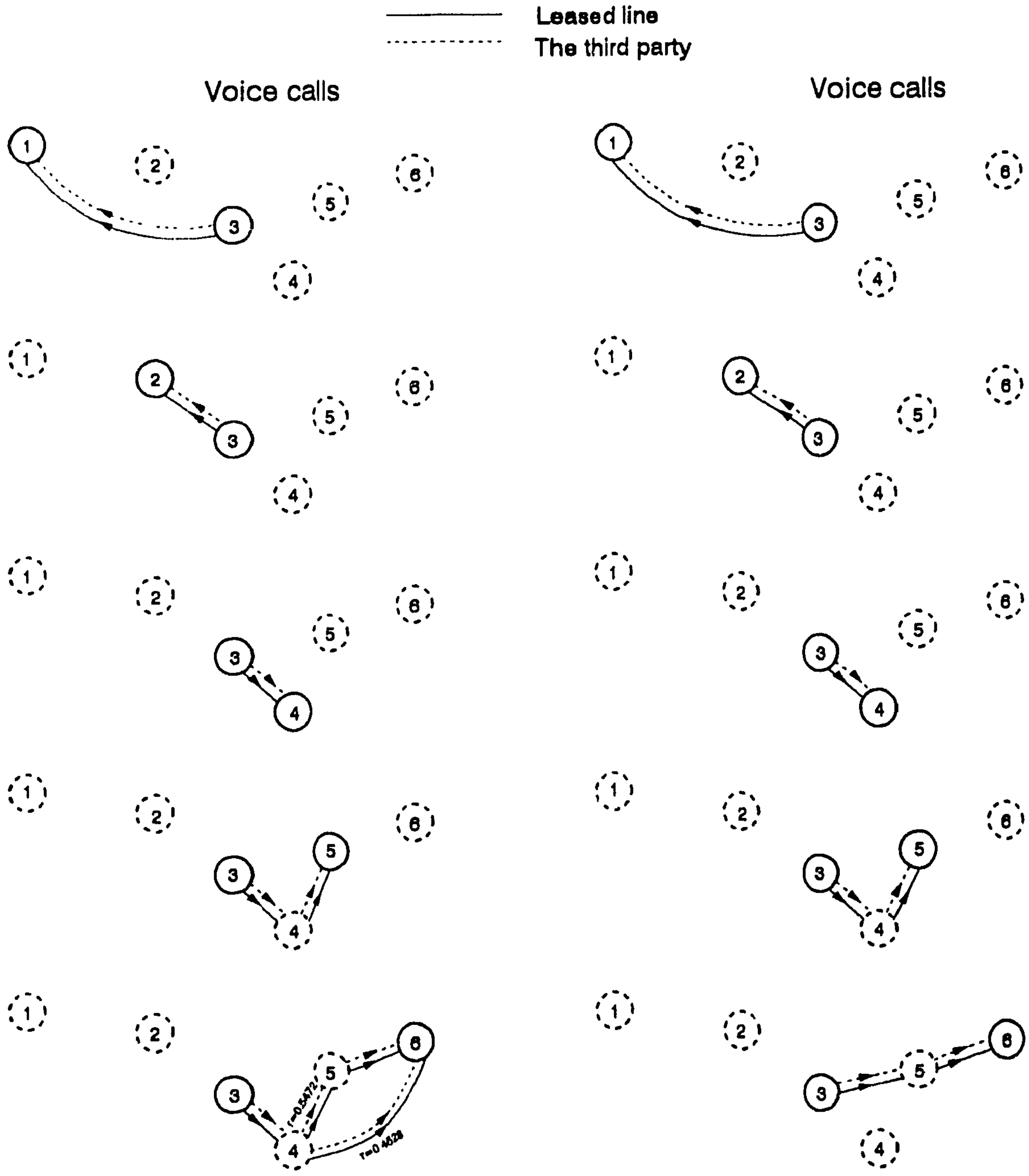


Fig. 3.3 The routes of the traffic for each O-D pair (cont'd)

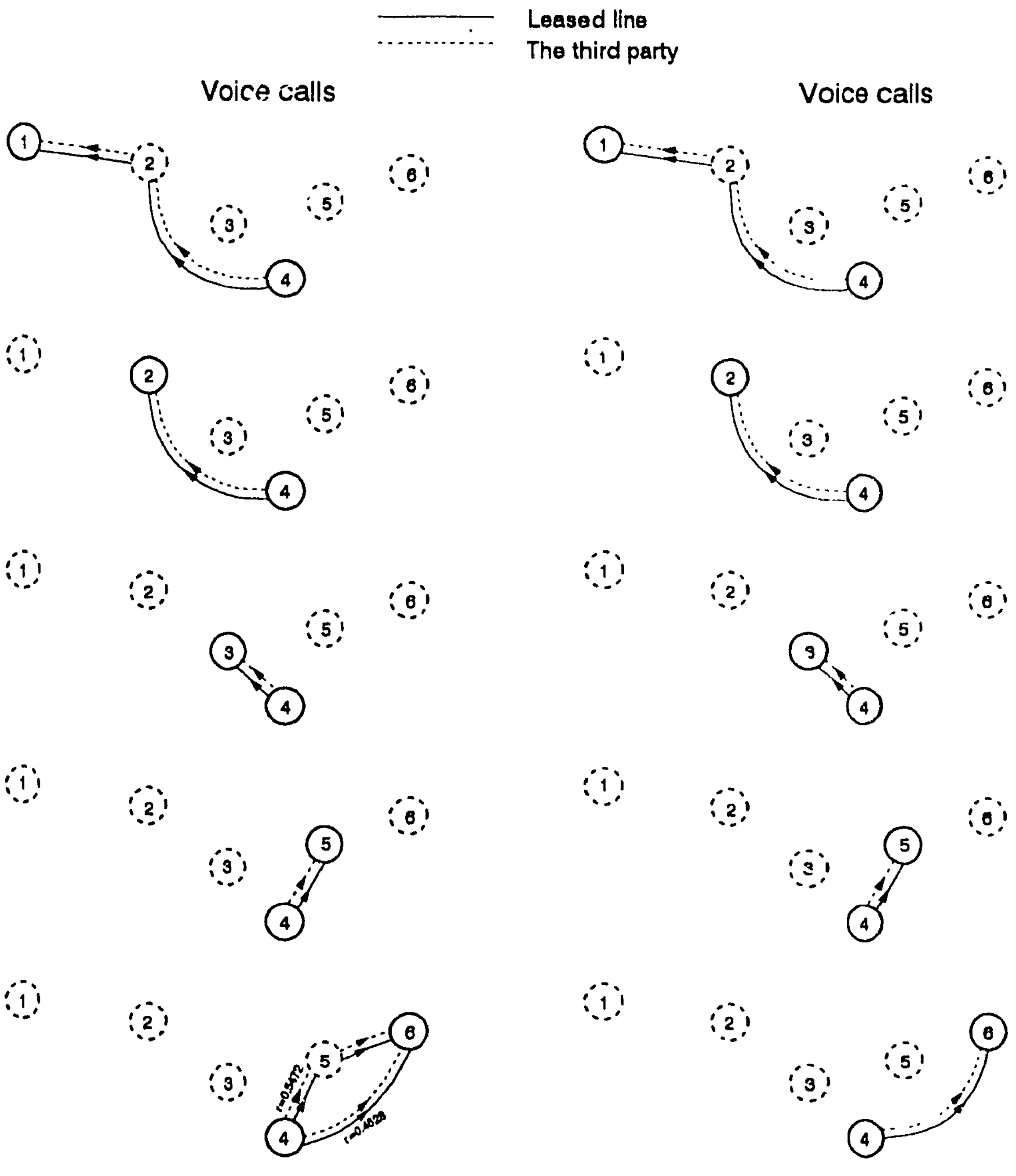


Fig. 3.3 The routes of the traffic for each O-D pair (cont'd)

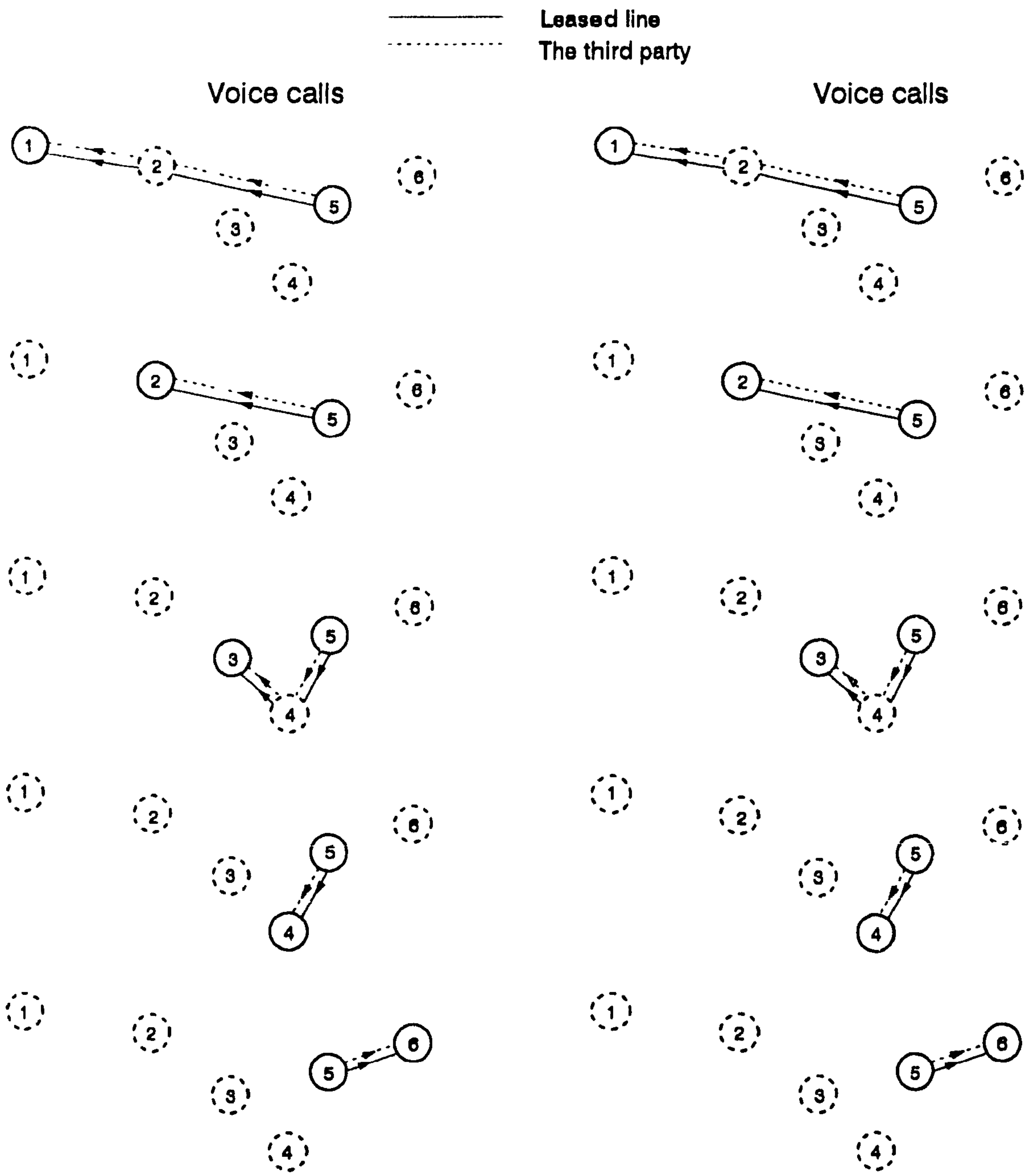


Fig. 3.3 The routes of the traffic for each O-D pair (cont'd)

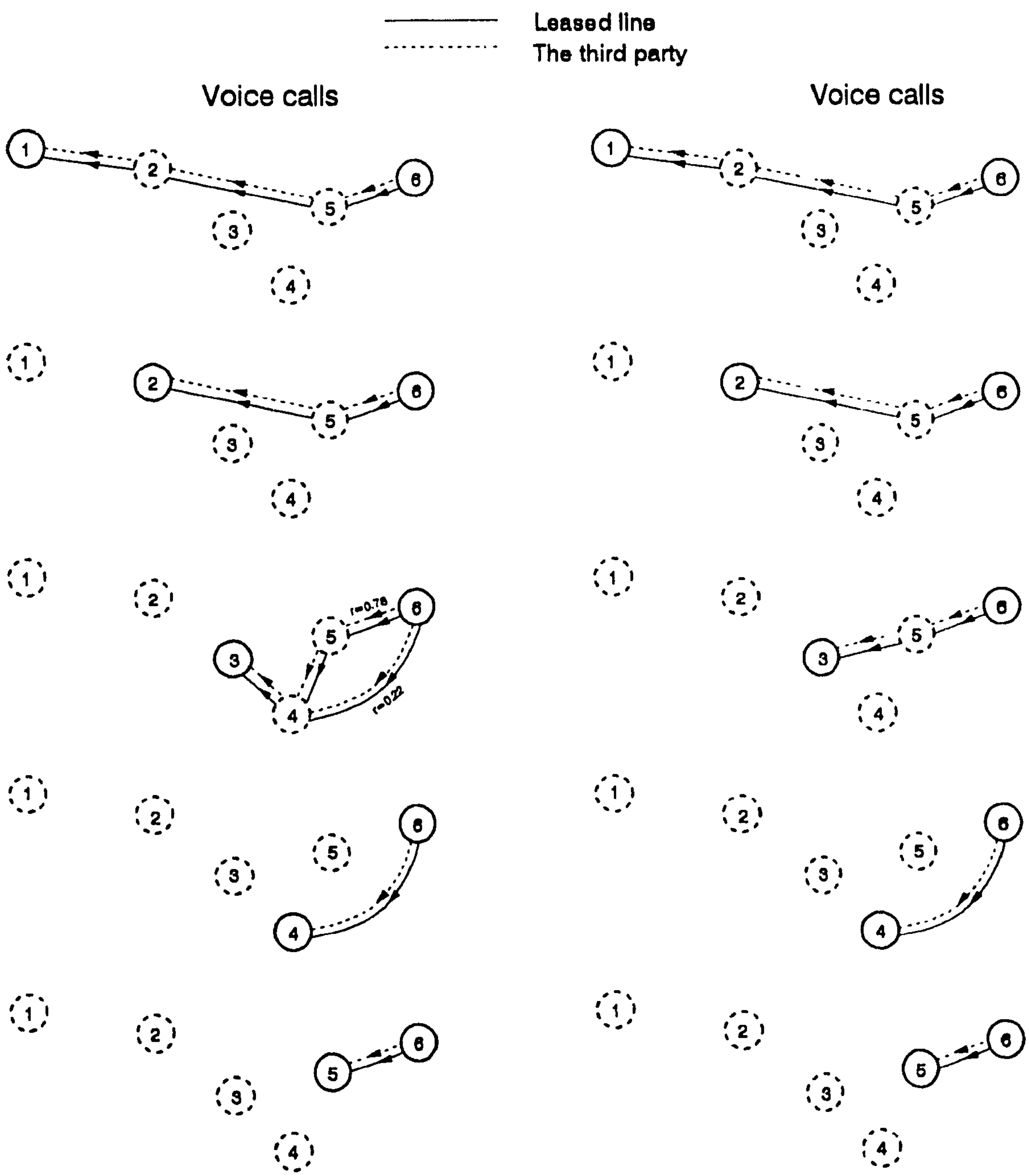


Fig. 3.3 The routes of the traffic for each O-D pair (cont'd)

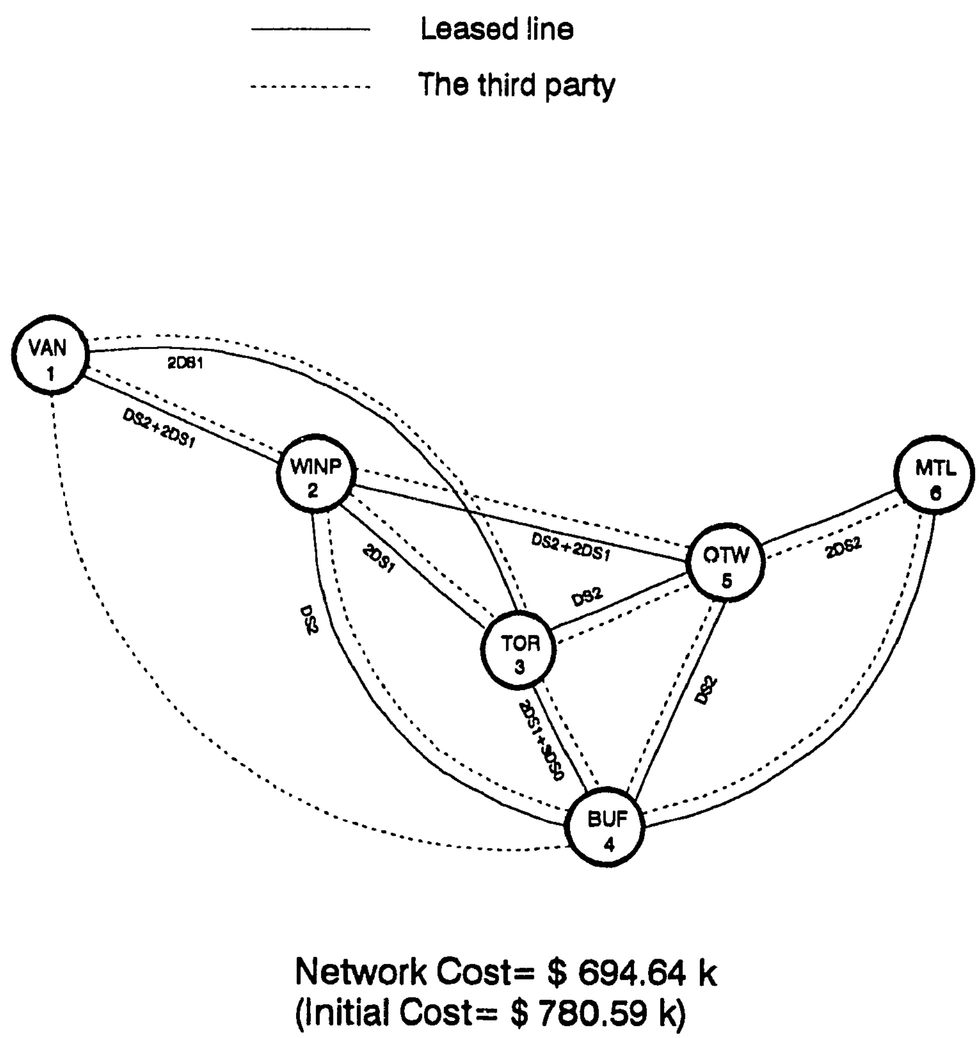


Fig. 3.4 The final physical topology of the 6-node network (2)

Table 3.6 Link flow of a 6-node network(2)

Link	Leased line		Third party	
	Voice (calls/min)	Video (calls/day)	Voice (calls/min)	Video (calls/day)
(1,2)	23.8801	13.1051	0.0986	2.8949
(1,3)	5.9249	2.6653	0.0751	1.3347
(1,4)	—	—	0.2103	—
(2,3)	5.9249	2.6653	0.0751	1.3347
(2,4)	11.9493	7.2033	0.0294	0.7967
(2,5)	23.9007	13.0967	0.0993	2.9033
(3,4)	5.9614	3.0483	0.0386	0.9517
(3,5)	11.9703	7.1980	0.0297	0.8020
(4,5)	11.9617	7.20	0.0296	0.7998
(4,6)	—	—	0.0087	0.0002
(5,6)	29.9537	16.8956	0.0376	1.1042

Table 3.7 Routing fraction for voice and video traffic of a 6-node network

Source	Destination	Link	Routing fraction		Source	Destination	Link	Routing fraction	
			Voice	Video				Voice	Video
1	2	(1,2)	1	1	2	1	(2,1)	1	1
		(1,3)	0	0			(2,3)	0	0
		(1,4)	0	0			(2,4)	0	0
		(1,5)	0	0			(2,5)	0	0
		(1,6)	0	0			(2,6)	0	0
	3	(1,2)	0	0		3	(2,1)	0	0
		(1,3)	1	1			(2,3)	1	1
		(1,4)	0	0			(2,4)	0	0
		(1,5)	0	0			(2,5)	0	0
		(1,6)	0	0			(2,6)	0	0
	4	(1,2)	0 9930	1		4	(2,1)	0	0
		(1,3)	0	0			(2,3)	0	0
		(1,4)	0 0070	0			(2,4)	1	1
		(1,5)	0	0			(2,5)	0	0
		(1,6)	0	0			(2,6)	0	0
	5	(1,2)	1	1		5	(2,1)	0	0
		(1,3)	0	0			(2,3)	0	0
		(1,4)	0	0			(2,4)	0	0
		(1,5)	0	0			(2,5)	1	1
		(1,6)	0	0			(2,6)	0	0
	6	(1,2)	1	1		6	(2,1)	0	0
		(1,3)	0	0			(2,3)	0	0
		(1,4)	0	0			(2,4)	0	0
		(1,5)	0	0			(2,5)	1	1
		(1,6)	0	0			(2,6)	0	0

Table 3.7 Routing fraction for voice and video traffic of a 6-node network (cont'd)

Source	Destination	Link	Routing fraction		Source	Destination	Link	Routing fraction	
			Voice	Video				Voice	Video
		(3,1)	1	1			(4,1)	0 0001	0
		(3,2)	0	0			(4,2)	0 9999	1
	1	(3,4)	0	0		1	(4,3)	0	0
		(3,5)	0	0			(4,5)	0	0
		(3,6)	0	0			(4,6)	0	0
		(3,1)	0	0			(4,1)	0	
		(3,2)	1	1			(4,2)	1	1
	2	(3,4)	0	0		2	(4,3)	0	0
		(3,5)	0	0			(4,5)	0	0
		(3,6)	0	0			(4,6)	0	0
		(3,1)	0	0			(4,1)	0	0
		(3,2)	0	0			(4,2)	0	0
3	4	(3,4)	1	1	4	3	(4,3)	1	1
		(3,5)	0	0			(4,5)	0	0
		(3,6)	0	0			(4,6)	0	0
		(3,1)	0	0			(4,1)	0	0
		(3,2)	0	0			(4,2)	0	0
	5	(3,4)	0	0		5	(4,3)	0	0
		(3,5)	1	1			(4,5)	1	1
		(3,6)	0	0			(4,6)	0	0
		(3,1)	0	0			(4,1)	0	0
		(3,2)	0	0			(4,2)	0	0
	6	(3,4)	0	0		6	(4,3)	0	0
		(3,5)	1	1			(4,5)	0 9988	0 9999
		(3,6)	0	0			(4,6)	0 0012	0 0001

Table 3.7 Routing fraction for voice and video traffic of a 6-node network (cont'd)

Source	Destination	Link	Routing fraction		Source	Destination	Link	Routing fraction	
			Voice	Video				Voice	Video
5	1	(5,1)	0	0	6	1	(6,1)	0	0
		(5,2)	1	1			(6,2)	0	0
		(5,3)	0	0			(6,3)	0	0
		(5,4)	0	0			(6,4)	0	0
		(5,6)	0	0			(6,5)	1	1
	2	(5,1)	0	0		2	(6,1)	0	0
		(5,2)	1	1			(6,2)	0	0
		(5,3)	0	0			(6,3)	0	0
		(5,4)	0	0			(6,4)	0	0
		(5,6)	0	0			(6,5)	1	1
	3	(5,1)	0	0		3	(6,1)	0	0
		(5,2)	0	0			(6,2)	0	0
		(5,3)	1	1			(6,3)	0	0
		(5,4)	0	0			(6,4)	0	0
		(5,6)	0	0			(6,5)	1	1
	4	(5,1)	0	0		4	(6,1)	0	0
		(5,2)	0	0			(6,2)	0	0
		(5,3)	0	0			(6,3)	0	0
		(5,4)	1	1			(6,4)	0 0017	0
		(5,6)	0	0			(6,5)	0.9983	1
	6	(5,1)	0	0		5	(6,1)	0	0
		(5,2)	0	0			(6,2)	0	0
		(5,3)	0	0			(6,3)	0	0
		(5,4)	0	0			(6,4)	0	0
		(5,6)	1	1			(6,5)	1	1

Chapter 4

MULTIPOINT ROUTING AND DIMENSIONING PROBLEMS IN CIRCUIT-SWITCHED NETWORKS

4.1 INTRODUCTION

Recall the situation of routing in point-to-point networks that at the time a point-to-point connection is established, one attempts to find the shortest available path connecting the desired pair of points. Nodes represent switching systems, edges represent links, the routing connections in point-to-point networks are treated as a shortest path problem in a graph. Routing a point-to-multipoint connection is comparable. Instead of the shortest path, one is interested in the shortest subtree of the network containing a given set of destination points. The problem is much more complex than the point-to-point routing problem. In fact, finding the shortest subtree connecting a set of destination points is a classical problem in graph theory (the Steiner tree problem) and is known to be NP-complete [25].

Several solutions have been proposed to address the multipoint routing problem for data networks. When a message (or packet) arrives at the network, the point-to-multipoint routing algorithm will choose one route so as to optimize a certain objective function. As far as the objective function is concerned, existing algorithms can be classified into the *Shortest-Path Algorithms* and the *Minimum-Cost Algorithms.*

Shortest-Path Algorithm

Given a set of destination nodes, the routes are computed independently from the source to each destination, using the shortest path; the paths are then merged into a single tree. This algorithm is used in [26] and [27]. Such a policy is clearly naive because the same message might travel several times over a certain link.

Minimum-Cost Algorithms

Given a set of destination nodes, the routing is done by constructing a tree that spans all destination nodes and sending the message along the links of this tree so as to minimize the sum of the costs of the links used. Many exact solutions and heuristics have been proposed for this well-known NP–complete problem. A comprehensive survey of the field is given in [48]. An important heuristic solution is the *Minimum Spanning Tree (MST) Heuristic* [25][28]. Kompella et al [29] proposed a variation of the MST heuristic to compute minimum-cost routes with a delay constraint. Another approach is to preselect a number of trees for a given network and then pose an optimization problem using these trees, as suggested by Bertsekas and Gallager [24] for the broadcast problem.

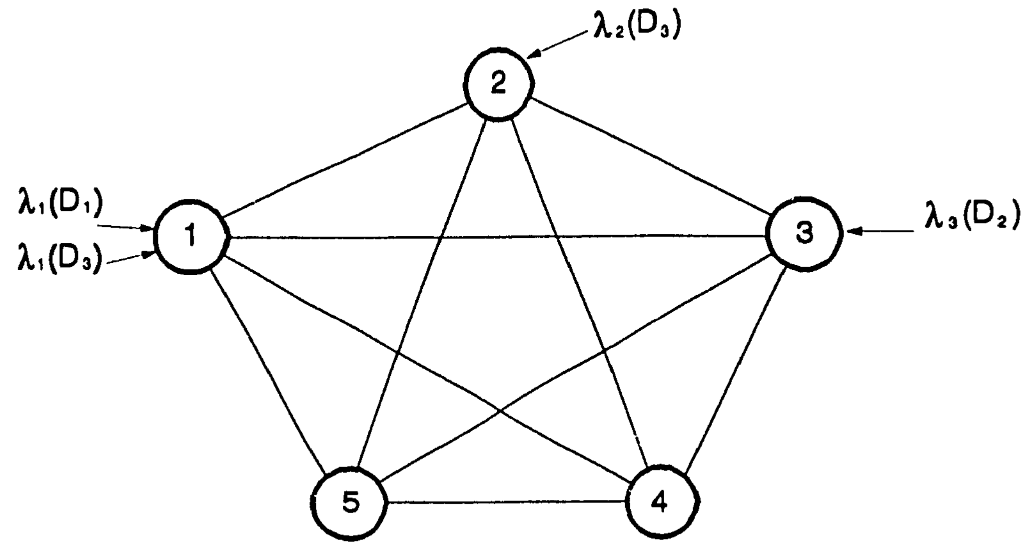
The algorithms mentioned above are able to route only a single source and destination set with fixed link costs. To compute the routes to handle a group of point-to-multipoint traffic, these algorithms have to be applied sequentially call by call. There is a scheme reported recently in [30] that presented an integer programming formulation for the problem of optimally routing point-to-multipoint streams.

We focus on the issue of determining optimal routes for multipoint connections in circuit-switched networks, under the assumption that the third party needs only to provide point-to-point capacity for the network to support multipoint services. Our approach is actually an extension of the flow-deviation method for point-to-point traffic, and employs heuristics to solve the problem approximately. A cost function can be specified and the multipoint optimization problem can be posed in a similar framework as described in Chapter 2. In the rest of the chapter, we will formulate the problem and develop two varieties of solution heuristics for the optimization. The first heuristic, the Minimum Cost Tree Heuristic (**MCTH**), serves as a benchmark for comparison purposes. The second heuristic is a modification of the preselection method suggested by Bertsekas and Gallager [24], and is called the Selected Minimum Cost Tree (**SMCT**) Heuristic. The heuristics were tested in experiments with a network of up to 20 nodes and evaluated by comparing the best network costs produced by them.

4.2 THE MODEL

The network has N nodes interconnected by bidirectional links. Assume that for an arbitrary number of source and destination-set pairs (i, D_j), calls arrive according to a stationary Poisson process with a mean arrival rate $\lambda_i(D_j)$, $i, j = 1, 2, \ldots$. Bandwidth along a route acquired at the start of a point-to-multipoint call, is reserved for its duration and released when the call terminates. All call holding times are independent and exponentially distributed with the same mean $1/\mu_m$. Let us look at a simple 5-node network illustrated in Fig. 4.1. There are four source and destination-set pairs $(1, \{2,3\}), (1, \{3,4\}), (2, \{3,4\}), (3, \{2,4\})$ with identical arrival rate of 2 calls/hour. Let $D_1 = \{2,3\}$, $D_2 = \{2,4\}$ and $D_3 = \{3,4\}$.

The traffic can be expressed in the form of a 5×3 matrix, its (i,j) element is $\lambda_i(D_j)$ if i is a source for a destination set D_j. Generally, we define $\lambda_l(D_j) = 0$ if l is not a source for a destination set D_j.



Original traffic matrix

$\lambda_i(D_j)$	$D_1 = \{2,3\}$	$D_2 = \{2,4\}$	$D_3 = \{3,4\}$
1	2	0	2
2	0	0	2
3	0	2	0
4	0	0	0
5	0	0	0

Fig. 4.1 Traffic inputs in a network

A call initiated by a source node will traverse a tree that connects to a set of destination nodes. Under the assumption that any multipoint call offered to the network is either transported over the leased lines or carried by the third party, the multipoint optimization problem can be posed in a similar way as a point-to-point optimization problem. The objective for routing point-to-point traffic is to divide each $\lambda_i(j)$ among the many paths from source i to destination j in a way that the resulting link flow pattern minimizes the total cost of network. The objective for point-to-multipoint routing is then to find all choices of routing fractions that divide $\lambda_i(D_j)$ among the many trees connecting source i to destination set D_j to minimize the total cost of network. The resulting link loads, as the function of routing fractions and arrival rates, determine a least-cost capacity assigned on each link.

A single type of traffic (voice or video) is assumed for all source and destination-set pairs for simplicity. It is possible to get the cost function of each link $C_{ij}(\rho_{ij}, d_{ij}, c^*_{ij}, o_{ij})$ in the same way as described in the previous chapters (Eq. (2.6) or Eq. (3.9)).

4.3 THE SOLUTIONS

Having described the model, we now turn to discuss the various heuristic solutions for it.

4.3.1 The Minimum Cost Tree Heuristic (MCTH)

In order to handle multipoint traffic, we need additional notation. Let a set of flow variables $\Lambda_i(D_j)$ denote the total calls at node i that are destined for a destination set D_j. The multipoint calls $\Lambda_i(D_j)$ will be routed over a set of trees $\{T_k\}$ of the network connecting i to all nodes in D_j. In fact, k is a notational number rather than true sequence. Let $V_{iD_j}^k$ be the set of nodes adjacent to node i in T_k, and $|V_{iD_j}^k|$ be the number of nodes in $V_{iD_j}^k$. Links $(i,l_1),(i,l_2),...,(i,l_k)$ with $l_1,l_2,...,l_k \in V_i^k$ are branches emanating from i in T_k. We use sets of routing variables $R_{iV_{iD_j}^k}(D_j)$ indicating the fraction of total traffic at node i destined for the destination set D_j that is routed along branches $(i,l_1),(i,l_2),...,(i,l_k)$ on tree T_k, and $r_{il}(D_j)$ indicating the fraction of traffic $\Lambda_i(D_j)$ that is routed over link (i,l). Then we have

$$\sum_k R_{iV_{iD_j}^k}(D_j) = 1 \tag{4.1}$$

and

$$r_{il}(D) = R_{iV_{iD_j}^k}(D_j), \quad \forall l \in V_{iD_j}^k \tag{4.2}$$

Fig. 4.2 gives an example of the relationship between routing variables $r_{il}(D_j)$ and $R_{iV_{iD_j}^k}(D_j)$ for the network shown in Fig. 4.1. Link $(1,2)$ is a branch on tree 1 ($V_{1D_3}^1 = \{2\}$), links $(1,3)$ and $(1,4)$ are branches on tree 2 ($V_{1D_3}^2 = \{3,4\}$), and $(1,5)$ is a branch on tree 3 ($V_{1D_3}^3 = \{5\}$). The fractions of traffic $\lambda_1(D_3)$ routed over branches on tree 1, tree 2 and tree 3 are 0.1, 0.2 and 0.7 respectively. Note that the sum of fractions of $\Lambda_i(D)$ routed over three trees must be equal to 1.

Since the traffic $\Lambda_i(D_j)$ at node i includes both input multipoint calls $\lambda_i(D_j)$ and the calls from other nodes that are routed through i for destination set D_j, we have

$$\Lambda_i(D_j) = \lambda_i(D_j) + \sum_{l,i\in V_{lD_j}^k \& |V_{lD_j}^k|=1} \Lambda_l(D_j) R_{lV_{lD_j}^k}(D_j), \quad \text{for } i,\ D_j \tag{4.3}$$

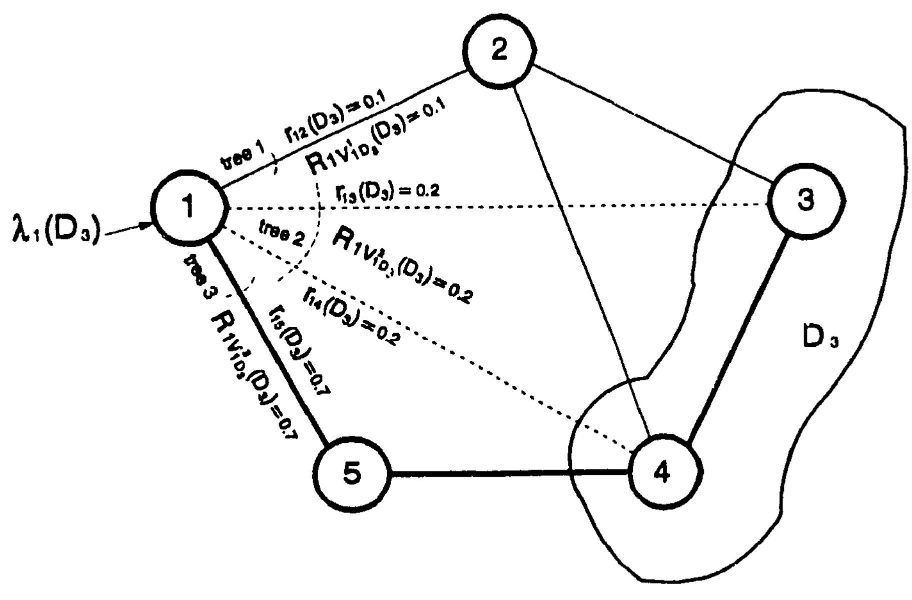


Fig. 4.2 The relationship between routing variables $r_{il}(D_j)$ and $R_{iV^k_{iD}}(D_j)$

Later, we will present the equation of the flow conservation at each node when $|V^k_{iD_j}| \geq 2$. Looking back to the example of Fig. 4.2, the total traffic at node 2 to D_3 becomes $\Lambda_2(D_3) = \lambda_2(D_3) + 0.1\lambda_1(D_3) = 2 + 0.1 \times 2 = 2.2$ calls/hour; and $\Lambda_5(D_3) = \lambda_5(D_3) + 0.7\lambda_1(D_3) = 0 + 0.7 \times 2 = 1.4$ calls/hour.

Traffic load on each link (i, j) is given by

$$\rho_{ij} = [\sum_{D_l, i \notin D_l} \Lambda_i(D_l) r_{ij}(D_l) + \sum_{D_l, j \notin D_l} \Lambda_j(D_l) r_{ji}(D_l)] / \mu_m \tag{4.4}$$

We then compute the cost of link (i, j) by Eq.(2.6). The differential cost of each link is

$$D_{ij} = \frac{\partial C_{ij}}{\partial \rho_{ij}} \tag{4.5}$$

Similar to the routing for point-to-point traffic, which optimizes the objective function if it routes traffic exclusively along paths of minimum derivative length, an optimal routing for multipoint traffic is to route traffic along trees with minimum differential cost. Initially, the network is assumed to be completely connected. All multipoint traffic is routed over the links connecting sources to each of the nodes in the destination sets, the initial routing variables are set as

$$R_{iD_j}(D_j) = \begin{cases} 1 & i \notin D_j \\ 0 & otherwise \end{cases} \tag{4.6}$$

for all i, D_j.

It follows that

$$r_{il}(D_j) = \begin{cases} 1 & l \neq i, \quad l \in D_j \\ 0 & otherwise \end{cases} \tag{4.7}$$

and

$$\Lambda_i(D_j) = \lambda_i(D_j)\,, \qquad all\ i,\ D_j \tag{4.8}$$

Fig. 4.3 shows an example of the initial traffic allocation. The input at node 1 destined for the destination set $D_3 = \{3,4\}$ goes on link $(1,3)$ and $(1,4)$.

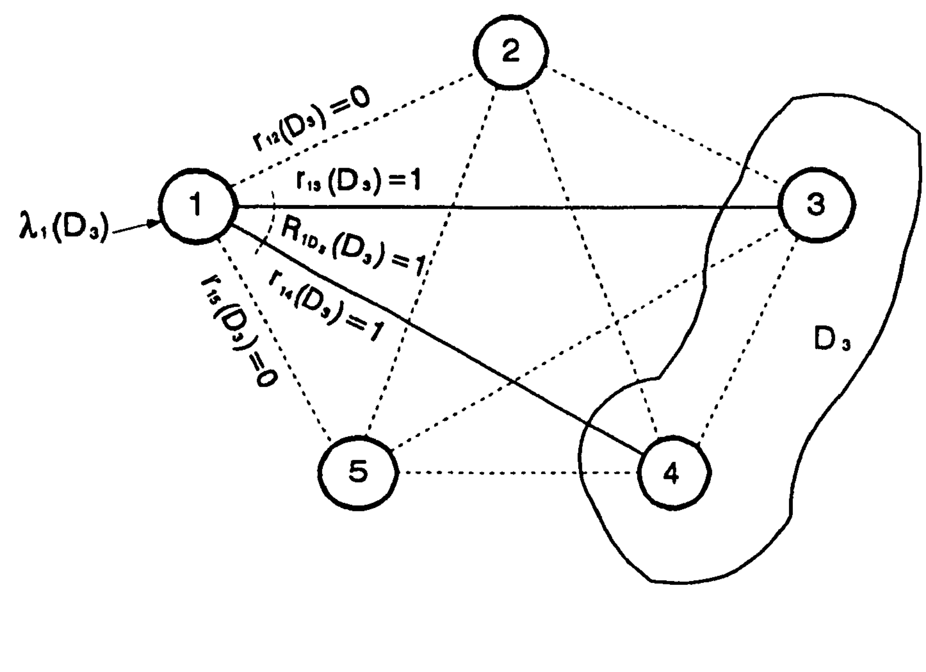


Fig. 4.3 Initial traffic flow allocation

Thereafter, we enter a sequence of iterations in which the algorithm involves two procedures:

1. **MSTH**: for computing an approximation to a least cost tree using Minimum Spanning Tree heuristic presented in [25]. The least cost tree, i.e., the tree with minimal sum of the costs of its edges, is called a Steiner tree.
2. **Value**: for updating routing variables, and corresponding node flows, link loads and differential costs.

MSTH

We now have a point-to-point network represented by $G = (V, E)$, where V is the node set and E is the edge set. The **MSTH** will find a near-optimal Steiner tree connecting a source node i to a set of destinations D_j. We briefly describe the MST heuristic, and demonstrate it on the graph in Fig. 4.4. The set of nodes to be spanned is $\{j, m\}$, and i is the source.

(1) Construct a closure graph G' on the set $\{i\} \cup D_j = \{i, j, m\}$. The closure is a complete graph on the nodes in $\{i\} \cup D_j$, with the cost of an edge (i, j) in this graph being the cost of the shortest path between i and j in G (Fig.4.4(a)). The costs of the shortest path are shown along the edges of the closure graph (Fig.4.4(b)).

(2) Determine a minimum spanning tree T of G' (Fig. 4.4(c)), based on Kruskal's algorithm [50].

(3) Finally, the edges in T are expanded to the edges in G that make up the least cost paths of G' (Fig. 4.4(d)).

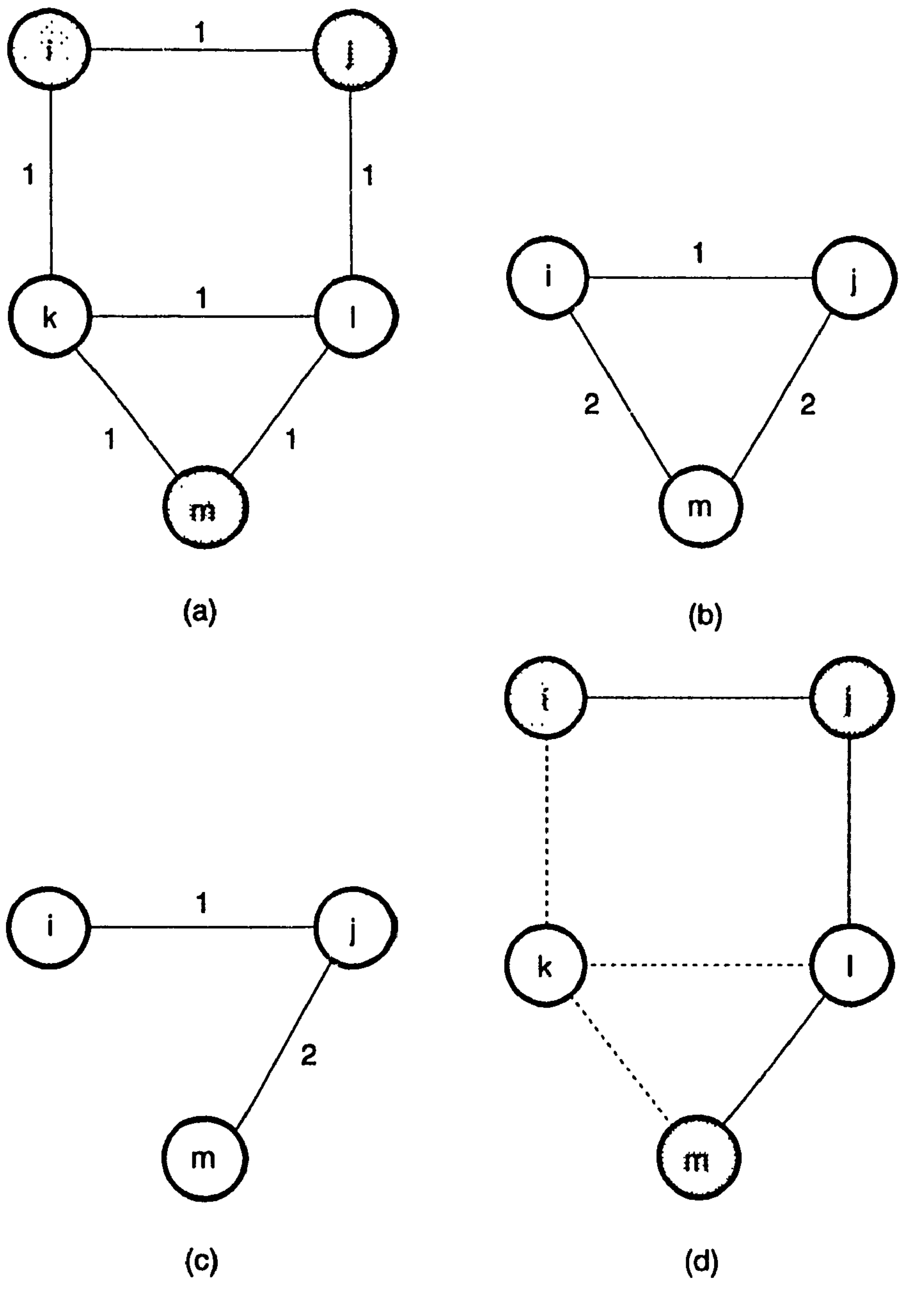


Fig. 4.4 The MST heuristic. (a) The graph (b) The closure (c) The MST (d) The Steiner tree

Value

The procedure **Value** adjusts routing variables by reducing the fractions allocated to previously-computed trees and increasing the fraction to be allocated to the newly-computed tree, in such a way as to maintain (4.1) in the process. As a result, node flows, link loads and differential costs will be updated. The procedure performs in accordance with the following rules:

(1) let $MST_i(D_j)$ denote the set of nodes adjacent to node i in the newly-computed Steiner tree connecting i to a set of destinations D_j. The new routing variable set at node i for destination D_j is regulated by

$$R_{iV_{iD_j}^k}(D_j) = \begin{cases} R_{iV_{iD_j}^k}(D_j) - \varepsilon_{iV_{iD_j}^k} & \text{if } V_{iD_j}^k \neq MST_i(D_j) \\ R_{iV_{iD_j}^k}(D_j) + \sum_{V_{iD_j}^l \neq MST_i(D_j)} \varepsilon_{iV_{iD_j}^l} & \text{if } V_{iD_j}^k = MST_i(D_j) \end{cases} \tag{4.9}$$

and

$$r_{il}(D_j) = \begin{cases} r_{il}(D_j) - \varepsilon_{iV_{iD_j}^k} & \forall l \in V_{iD_j}^k, V_{iD_j}^k \neq MST_i(D_j) \\ r_{il}(D_j) + \sum_{V_{iD_j}^l \neq MST_i(D_j)} \varepsilon_{iV_i^l} & \forall l \in V_i^k, V_i^k = MST_i(D) \end{cases} \tag{4.10}$$

where

$$\varepsilon_{iV_{iD_j}^k} = \eta * R_{iV_{iD_j}^k}(D_j), \quad \eta \in [0,1] \tag{4.11}$$

The value of η is determined by the line search method described in section 2.4.

Fig. 4.5 shows an example of routing variable set mapping. The solid links represent branches of a previous tree, the thick links denote a new Steiner tree, with $MST_i(D) = \{d_3, l\}$. We now see that the fraction of traffic at node i destined for $D = \{d_1, d_2, d_3\}$ on the link (i, d_3) keeps unchanged when link (i, d_3) belongs to both the previous tree and the new tree.

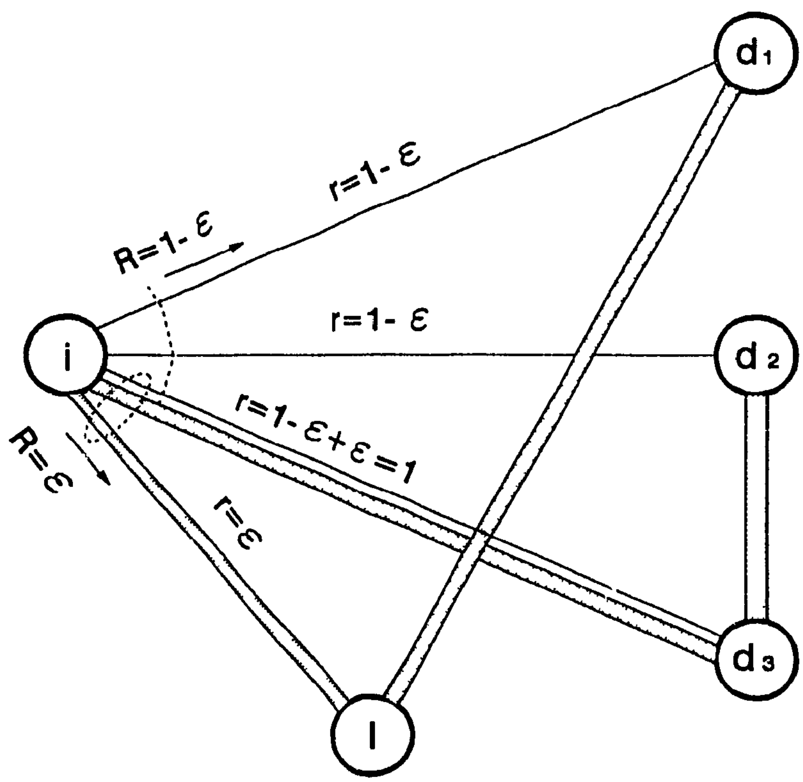


Fig. 4.5 Mapping of routing variables

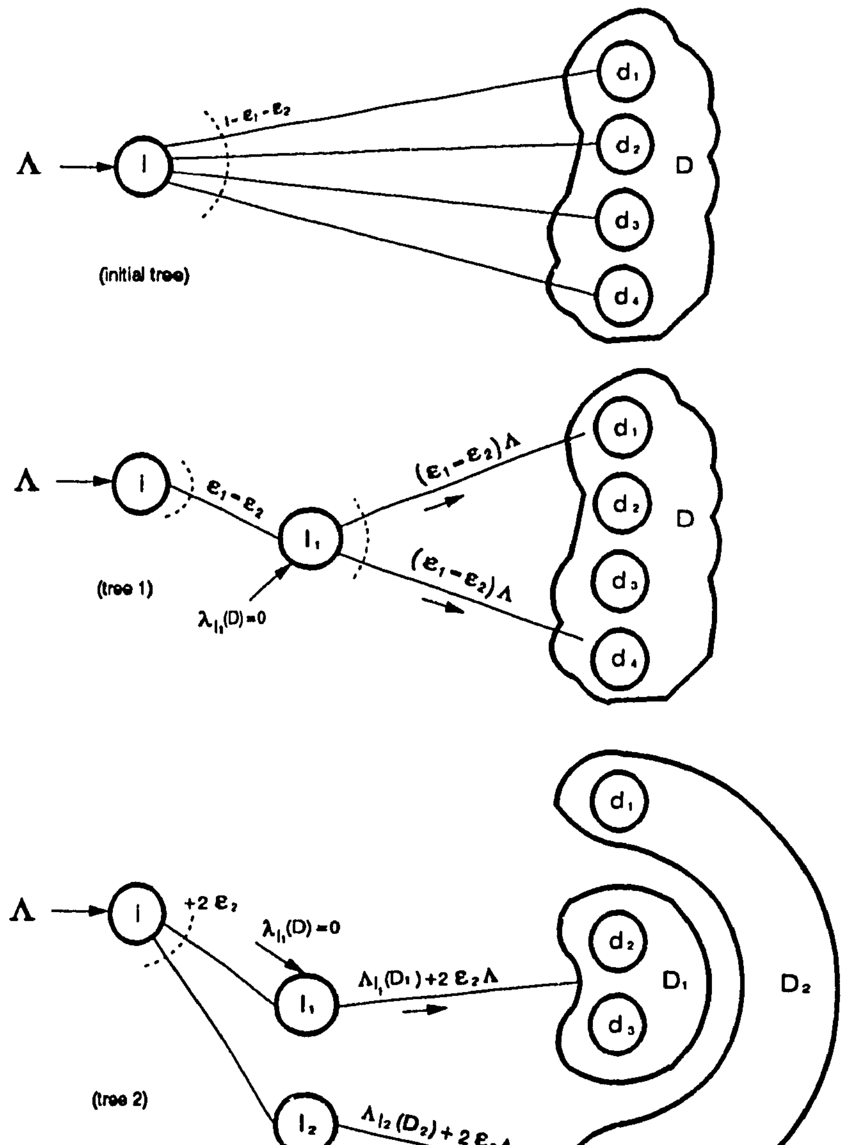


Fig. 4.6 Mapping of node flows

(2) For a source and destination-set pair (i, D_j), the destination set D_j may be partitioned into a number of smaller sets as the algorithm proceeds. Seen from an example shown in Fig. 4.6, the destination set D will be divided into two subsets. A few definitions are needed here. Let $D^k_{j,l}$ denote the subset of D_j reachable from node l in a tree T_k, where $l \in V^k_{iD_j}$ & $l \notin D_j$. We have $D_j = \cup_{l \in V^k_{iD_j}} D^k_{j,l}$. Also let $D^{MST}_{j,l}$ denote the subsets of D_j reachable from node l in the newly-computed tree. In the example of Fig. 4.6, the set of nodes adjacent to node i in tree 1 and tree 2 (newly-computed tree) are $V^1_{iD} = \{l_1\}$ and $V^2_{iD} = \{l_1, l_2\}$, $D^{MST}_{l_1} = D_1$ and $D^{MST}_{l_2} = D_2$ are the subsets of D reachable from node l_1 and l_2 in tree 2. We can see that $D_1 \cup D_2 = D$.

For each source i and each destination set D_j, we first update the total traffic at node $l \in V^k_{iD_j} \& |V^k_{iD_j}| = 1$ destined for D_j according to Eq.(4.3), i.e., $\Lambda_l(D_j) = \lambda_l(D_j) + \sum_{i, l \in V^k_{iD_j} \& |V^k_{iD_j}|=1} \Lambda_i(D_j) R_{iV^k_{iD_j}}(D_j)$. We then consider the case $|V^k_{iD_j}| \geq 2$. When $|V^k_{iD_j}| \geq 2$, the destination set D_j is partitioned into several subsets $D^k_{j,l}$. Without loss of generality, we specify a set $B_i(D_j) = V^k_{iD_j}$ $(|V^k_{iD_j}| \geq 2)$ of blocked nodes l for which $\Lambda_l(D_j)$ remains unchanged but $\Lambda_l(D^k_{j,l})$ is changed. The total traffic at each node $l \in V^k_{iD_j}$ destined for any subset of D_j can be computed according to

$$\Lambda_l(D^k_{j,l}) = \Lambda_l(D^k_{j,l}) + R_{iV^k_{iD_j}} \Lambda_i(D_j) \tag{4.12}$$

Returning to the example of Fig. 4.6, $l_1 \in V^1_{iD}$ & $|V^1_{iD}| = 1$, the total traffic at node l_1 to D is adjusted by $\Lambda_{l_1}(D) = \Lambda_{l_1}(D) + (\varepsilon_1 - \varepsilon_2)\Lambda$ at first; then the total traffic at node l_1 destined for D_1 and the total traffic at node l_2 destined for D_2 becomes $\Lambda_{l_1}(D_1) = \Lambda_{l_1}(D_1) + 2\varepsilon_2\Lambda$, and $\Lambda_{l_2}(D_2) = \Lambda_{l_2}(D_2) + 2\varepsilon_2\Lambda$; while $\Lambda_{l_1}(D)$ and $\Lambda_{l_2}(D)$ remain unchanged.

If the subset $D^k_{j,l}$ is a new destination set, i.e., $D^k_{j,l} \notin$ the original destination sets $D_1, ..., D_j, ...$ in the traffic matrix, we have to initialize the routing variables at each node m in the same way as (4.6) and (4.7):

$$R_{mD^k_{j,l}}(D^k_{j,l}) = \begin{cases} 1 & j \neq i \quad j \notin D^k_{j,l} \\ 0 & others \end{cases} \tag{4.13}$$

and

$$r_{mn}(D^k_l) \Longleftarrow R_{mD^k_{j,l}}(D^k_{j,l}), \quad \forall n \in D^k_{j,l} \tag{4.14}$$

If a node $l \in V^k_{iD_j}$ & $l \in D_j$, but $l \notin \{$ leaves in T_k $\}$, then a subset $D' \longleftarrow D \backslash l$ is created.

$$\Lambda_l(D') \Longleftarrow \Lambda_l(D_j) R_{iV^k_{iD_j}}(D_j) \tag{4.15}$$

Again, the routing variables will be specified as the same way as (4.13) and (4.14), if D' is a new destination set.

(3) The link loads and differential costs are updated by (4.4) and (4.5), respectively.

Cycling through the source destination-set pairs in successive iterations, each iteration processes a single source destination-set pair (i, D_j). The algorithm terminates when the routing variables remain unchanged after a round of iterations. The algorithm is listed in Fig. 4.7.

The main difficulty with the procedures of the algorithm is one of complexity. Some new destination sets may be emerged during the processing. However, for a small number of source destination-set pairs and small destination sets D_j, the number of subsets that result is manageable.

```
Initialization;
while (termination criteria are not met)
  do i ∈ {1,2,..,N}
    do D_j ∈ {D_1, D_2,..}
      if Λ_i(D_j) ≠ 0
        call MSTH (        );
        call Value(        );
        test termination condition;
      end if
    end do
  end do
```

Fig. 4.7 The sequence of the MCTH algorithm

4.3.2 Selected Minimum Cost Tree (SMCT) Heuristic

The **SMCT** heuristic is adapted from the preselection approach presented in [24] by Bertsekas and Gallager for the broadcast problems in data networks. Suppose that there is traffic Λ_i to be broadcast from node i. The trees to be used for routing are to be selected from a given collection $\{T_k\}$. The total flow of a link is the sum of all tree flows traversing this link. By expressing the total flow of a link in terms of tree flows, the routing problem can be formulated in terms of the unknown portions of Λ_i broadcast along trees in $\{T_k\}$ that minimize a total link cost. This problem can be solved by the flow-deviation method. Our approach uses this idea of routing the traffic flow over several trees and finding all choices of fractions of traffic flow on these trees so as to minimize the total cost of network. However, the trees are not preselected. We choose them from the trees arising as the algorithm progresses through its iterations.

We introduce new definitions:

$R_i^n(D_j)$: the fraction of multipoint calls (i, D_j) routed over the nth Steiner tree interconnecting i with its destination set D_j

$T_{iD_j}^n$: the $N \times N$ topology matrix of the nth Steiner tree of (i, D_j). Its (l, m) element $T_{iD_j}^n\big|_{lm}$ is 1 if link (l, m) is a branch of the tree, and 0 otherwise. For example, the topology matrix of tree 2 in Fig. 4.2 is

$$T_{1D_3}^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$DD_i^n(D_j)$: the differential cost of the nth Steiner tree of (i, D_j)

$\Delta_i^n(D_j)$: the amount of variation of $R_i^n(D_j)$

For any source and destination-set pair (i, D_j), we have

$$\sum_n R_i^n(D_j) = 1 \tag{4.16}$$

Assume k trees will be selected for each (i, D_j). First, all traffic is allocated on the trees merged by the direct links connecting sources to each nodes in the destination sets. The algorithm starts with

$$R_i^1(D_j) = \begin{cases} 1 & all \ \ (i, D_j) \quad \lambda_i(D_j) \neq 0 \\ 0 & others \end{cases} \tag{4.17}$$

$$R_i^2(D) = \cdots = R_i^k(D) = 0 \quad all \ \ (i, D_j) \tag{4.18}$$

The elements of $T^1_{iD_j}$ are set by

$$T^1_{iD_j}\Big|_{mn} = \begin{cases} 1 & if\ m = i\ \&\ n \in D_j \\ 0 & others \end{cases} \tag{4.19}$$

All elements of $T^2_{iD_j}, ..., T^k_{iD_j}$ are set to zero. The traffic load on link (i, j) is calculated by

$$\begin{aligned} \rho_{ij} &= \sum_{l,D_j} \lambda_l(D_j)[\sum_k R^k_l(D_j) T^k_{lD_j}\Big|_{ij}] + \sum_{l,D_j} \lambda_l(D_j)[\sum_k R^k_l(D_j) T^k_{lD_j}\Big|_{ji}] \\ &= \sum_{l,D_j} \lambda_l(D_j) \sum_k R^k_l(D_j)[T^k_{lD_j}\Big|_{ij} + T^k_{lD_j}\Big|_{ji}] \end{aligned} \tag{4.20}$$

The differential cost of each link is given by (4.5) and the total differential cost of the kth tree of (i, D_j) is

$$DD^k_i(D_j) = \sum_{l=1}^{N} \sum_{m=1}^{N} D_{lm} * T^k_i(D_j)\Big|_{lm}, \tag{4.21}$$

The general procedure, which processes a single (i, D_j) pair in each iteration, consists of the following steps:

(1) Find a new Steiner tree by the Minimum Spanning Tree heuristic [25].

(2) Compare the topology of this newly-computed tree with $T^1_{iD_j}, T^2_{iD_j}, ..., T^k_{iD_j}$.

If there is a tree $T^j_{iD_j}$ identical with this newly-computed tree, then for each of k trees, we compute

$$\Delta^l_i(D_j) = -\min\{R^l_i(D_j), \eta(1 - \frac{DD^j_i(D_j)}{DD^l_i(D_j)})\} \quad l \in \{1, ..., k\}, l \neq j \tag{4.22}$$

$$\Delta^j_i(D_j) = -\sum_{l=1, l\neq j}^{k} \Delta^l_i(D_j) \tag{4.23}$$

The scale factor η is determined by the line search method described in section 2.4. We update the routing fractions over the k trees

$$R_i^n(D_j) = R_i^n(D_j) + \Delta_i^n(D_j), \quad n = 0, 1, ..., k \tag{4.24}$$

Otherwise, find a $T_{iD_j}^j$ in k trees with minimum routing fraction (i.e., $R_i^j(D_j) = \min\{R_i^1(D_j), ..., R_i^k(D_j)\}$), and then

(a) $T_{iD_j}^j \Longleftarrow$ topology of newly-computed tree

(b) Check the new cost of network. If reduction is obtained over the network cost, we set $DD_i^j(D_j) \Longleftarrow$ differential cost of the newly-computed tree, and update routing fractions over the k trees by (4.22), (4.23) and (4.24). If no reduction is obtained, ignore this newly-computed tree and find a $T_{iD_j}^j$ in k trees with minimum differential cost (i.e. $DD_i^j(D_j) = \min\{DD_i^1(D_j), ..., DD_i^k(D_j)\}$). Again update routing fractions by (4.22), (4.23) and (4.24).

(3) Re-calculate link loads by (4.20) and the differential cost of the k trees by (4.21).

The algorithm terminates when routing fractions over the k trees and topology of the trees remain unchanged. The sequence of the algorithm is listed in Fig. 4.8.

4.4 NUMERICAL RESULTS

In this section we discuss some numerical examples that serve to illustrate various aspects of the performance of our proposed heuristics. The external traffic

```
Initialization;
while (termination criteria are not met)
   do i ∈ {1,2,..,N}
      do D_j ∈ {D_1, D_2, ..}
         if λ_i(D_j) ≠ 0
            Find a new Steiner tree;
            Update routing fractions over k trees;
            Update link loads and differential costs;
            test termination condition;
         end if
      end do
   end do
```

Fig. 4.8 The sequence of the SMCT algorithm consists of point-to-point voice calls and multipoint voice calls (such as voice conference). Point-to-point calls will be routed first. We will begin to route multipoint calls when the algorithm for point-to-point voice traffic routing reaches its steady state. The traffic load on each link in a network is simply the sum of point-to-point and multipoint traffic routed over this link. The call holding time is assumed to be 3.5 mins for point-to-point voice calls, and 30 mins for multipoint voice calls. The number of nodes in the networks is 20 in all cases. The original traffic matrix and the final flows allocated on each link for point-to-point voice calls are presented in Table 2.12 and Table 2.14 respectively.

First we apply our heuristics for routing a single multipoint call with arrival rate $\lambda_5(9, 18, 19) = 2$ calls/hour (source node: 5, destinations: $\{9, 18, 19\}$). We choose $k = 5$ in the **SMCT** heuristic. The results of the two algorithms are exactly same. Fig. 4.9 shows the tree obtained by the algorithms. There is 10.56% decrease in the cost of network. We then test the two heuristics by routing a single multipoint call or 2 multipoint calls with various sources and destination sets (k=3 for all source destination-set pairs). The corresponding results are summarized in Table 4.1 and Table 4.2. The results have shown that the **SMCT** algorithm performs reasonably well, producing multipoint connections having network costs that are 0.02% within the cost of **MCTH**. The **SMCT** algorithm, as expected, runs much faster than the **MCTH** algorithm. For a single 10-destination multipoint call, the run time of the **SMCT** is less by a factor of 10 or more, compared with the run time of **MCTH** algorithm.

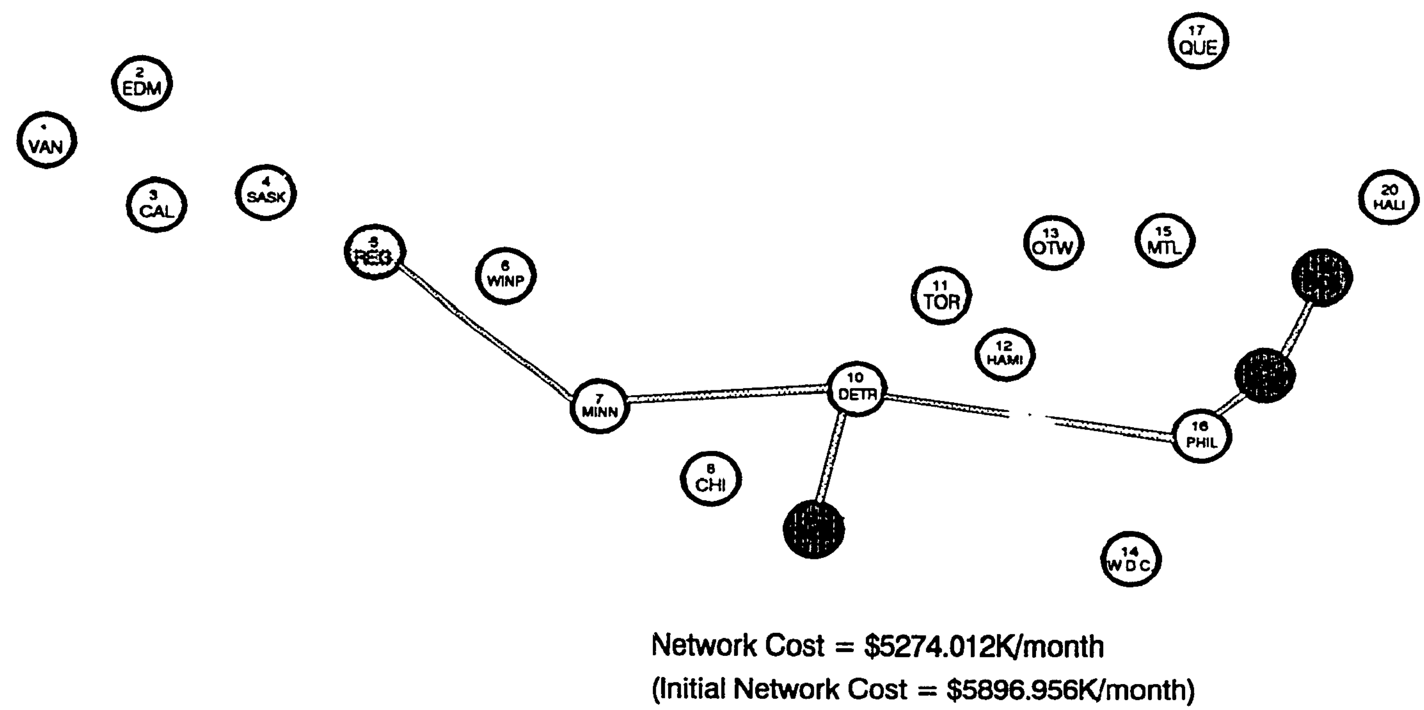


Fig. 4.9 An example of a single multipoint routing

Table 4.1 Comparison of two heuristic solutions (single multipoint call)

Source	Destinations	Arrival rate (calls/hour)	Network cost ($/month)		
			Initial	MCTH	SMCT ($k = 3$)
1	{6,8,10,11}	2	5901.492K	5268.064K	5268.064K
7	{8,10,14,16,18}	3	5917.368K	5283.141K	5283.141K
8	{7,10,11,16,18,19}	4	5937.780K	5302.241K	5302.428K
10	{7,8,9,14,16,18,19}	2	5915.1K	5284.368K	5284.974K
11	{1,5,6,8,10, 13,15,18,20}	4	5964.996K	5313.646K	5314.574K
18	{1,6,7,8,9,10, 11,14,16,19}	3	5951.388K	5312.709K	5313.684K

Table 4.2 Comparison of two heuristic solutions (two multipoint calls)

Source	Destinations	Arrival rate (calls/hour)	Network cost ($/month)		
			Initial	MCTH	SMCT ($k = 3$)
1	{10,12}	1			
9	{16,18}	2	5896.956K	5330.844K	5330.844K
6	{8,9,18}	3			
17	{14,16,18}	3	5964.996K	5356.664K	5356.665K
7	{6,10,16,18}	4			
14	{7,8,10,18}	5	6046.644K	5490.352K	5490.360K
10	{1,5,13,15,20}	1			
18	{8,9,14,16,19}	4	5996.748K	5427.057K	5427.432K

We have further tested the **SMCT** algorithm by routing ten multipoint calls, each with five destinations. Although the size of memory that stores the topology information of the k trees grows exponentially as the number of multipoint calls increases, our preliminary study shows that it is enough to consider four trees for each multipoint call to obtain a very close approximation of the optimum routing (see Table 4.3).

Table 4.3 Results of **SMCT** with different k

Source	Destinations	Arrival rate	Network cost (K$/month)				
		(calls/min)	$k = 10$	$k = 5$	$k = 4$	$k = 3$	$k = 2$
1	{3,6,11,12,13}	1					
1	{6,8,10,11,15}	1.5					
8	{7,9,10,16,18}	3					
8	{10,14,16,18,19}	2					
8	{10,11,16,18,20}	2					
10	{5,8,9,14,16}	2					
11	{1,2,6,13,15}	1					
11	{12,13,15,17,20}	1	7048.825	7048.831	7048.909	7049.47	7051.99
17	{8,10,11,14,16}	2					
20	{1,4,6,13,19}	1					

(Initial network cost = $7754.448K/month)

4.5 CONCLUDING REMARKS

Multipoint calls supporting services such as voice and video conference require circuits interconnecting all nodes in a subset of network nodes. For each multipoint call, there is a source (the originator) and a destination set (the called party). The interconnections for such a call can be realized by allocating routes along a tree of the network. Under the assumption that the third party needs only to provide point-to-point capacity for the network to support multipoint services, the multipoint optimization problem in circuit-switched network can then be posed in a similar framework as the optimization problems for point-to-point traffic.

In this chapter, two approximate solution methods for multipoint optimization were proposed: the Minimum Cost Tree Heuristic (**MCTH**) and the Minimum Spanning Tree Storing (**SMCT**) Heuristic. The heuristics were described and their performance was tested by routing various multipoint calls in a 20-node network. The experimental results showed that both heuristics converged to valid trees. The run time for the **SMCT** is an order or more of magnitude less than that for the **MCTH**. For large networks and a larger number of destinations in each source destination-set pair, the **MCTH** is not practical. Our emphasis is on developing practical algorithms for large, general purpose networks. In such an environment, an algorithm must not only provide good performance, but must lend itself to an efficient implementation. So the main use of **MCTH** is as benchmark for other multipoint optimization algorithms. The motivation underlying **SMCT** is that it often not worth the degree of effort that is needed to find the optimal solution when an approximation (which may be found quite easily) will provide sufficiently good performance. The attractive features of the **SMCT** algorithm are that it will obtain a reasonable result by properly choosing k, and it runs fast. Although the experimental testing was substantial and the results are encouraging, still future research is necessary to establish theoretical bounds on these heuristics.

Chapter 5

MULTICAST ROUTING FOR PACKET-SWITCHED NETWORKS

Multicasting is the simultaneous transmission of data to multiple destinations. The multicast routing algorithm is responsible for finding a route in the network that satisfies the traffic requirements, while optimizing a certain objective function. Previous optimization techniques for multicast routing in data networks have considered two optimization goals, the delay or the cost. The objective of this chapter is to modify the multipoint formulation for circuit-switched networks into a form applicable to the optimal routing problem for multicast packet-switched traffic, where the optimization goal is the total average delay in the network.

5.1 PROBLEM FORMULATION FOR MULTICAST ROUTING

5.1.1 The Network

The network is a collection of nodes, interconnected by links subject to a certain topology. We consider all links to be point-to-point and directed (i.e., information can flow in only one direction). Full-duplex connections between pairs of nodes correspond to two independent links, one in each direction. Each link is characterized by the capacity in data units/second. Formally, a network with N nodes and M links is denoted by $G(A, W)$, where A is the $N \times M$ topology matrix (its (i, j) element is 1 if node i is the origin of link j, -1 if it is the destination, and 0 otherwise) and W is the link capacity vector.

5.1.2 The traffic

The input arrivals are assumed according to a stationary Poisson process with rate $\lambda_i(D)$, which is the arrival rate of traffic entering the network at node i and destined for a set of destinations D (measured in data units/second). This multicast stream will be routed over a set of trees connecting i to all nodes in D. Let $R_i^n(D)$ be the fraction of the multicast traffic $\lambda_i(D)$ that is routed over the nth tree, and T_{iD}^n be the $N \times N$ topology matrix of this nth tree, its (l,m) element $T_{iD}^n|_{lm}$ is 1 if link (l,m) is a branch of the tree and 0 otherwise. The total flow f_{ij} on link (i,j) is computed according to

$$f_{ij} = \sum_{(l,D)} \lambda_l(D) \sum_n R_l^n(D) T_{lD}^n|_{ij} \tag{5.1}$$

5.1.3 The Cost Function

Under the Kleinrock [51] independence assumption, namely independence of service time at successive nodes, a communication network can be modeled as a network of independent $M/M/1$ queues. Under the assumption that an exponentially distributed amount of time is required to transmit a packet, the expression of the average network delay is given by the well known formula

$$\frac{1}{\gamma} \sum_{i,j} \frac{f_{ij}}{c_{ij} - f_{ij}} \tag{5.2}$$

where c_{ij} is the capacity of link (i,j), $0 \leq f_{ij} < c_{ij}$, and γ is the total arrival rate offered to the network. Even if the packets are not exponentially distributed, the convex function of the form given in (5.2) is a good measure of performance.

Generally speaking, mean optimum solutions are robust with respect to performance measures [52]. Since the total external load γ is constant, the cost function appropriate for optimization is in the form

$$\sum_{(i,j)} T_{ij}(f_{ij}) = \sum_{i,j} \frac{f_{ij}}{c_{ij} - f_{ij}} \tag{5.3}$$

which, from Little's formula, has the interpretation of the average number of packets in the network.

5.1.4 Problem Formulation

Our problem can now be stated as follows:

GIVEN:

— A packet-switched network with topology $G(A, W)$

— A set of stationary multicast traffic external arrival rates $\{\lambda_i(D)\}$

OBJECT:

$$\text{Minimize} \quad \sum_{i,j} T_{ij}(f_{ij}) \tag{5.4}$$

WITH RESPECT TO:

All choices of routing variables $\{R_i^n(D)\}$

UNDER THE CONSTRAINT:

$$\sum_n R_i^n(D) = 1, \quad \text{all} \quad (i, D) \tag{5.5}$$

5.2 THE ALGORITHM

The algorithm for this multicast case proceeds exactly as described in section 4.3.2 (SMCT heuristic), while the length of each link (i, j) is periodically

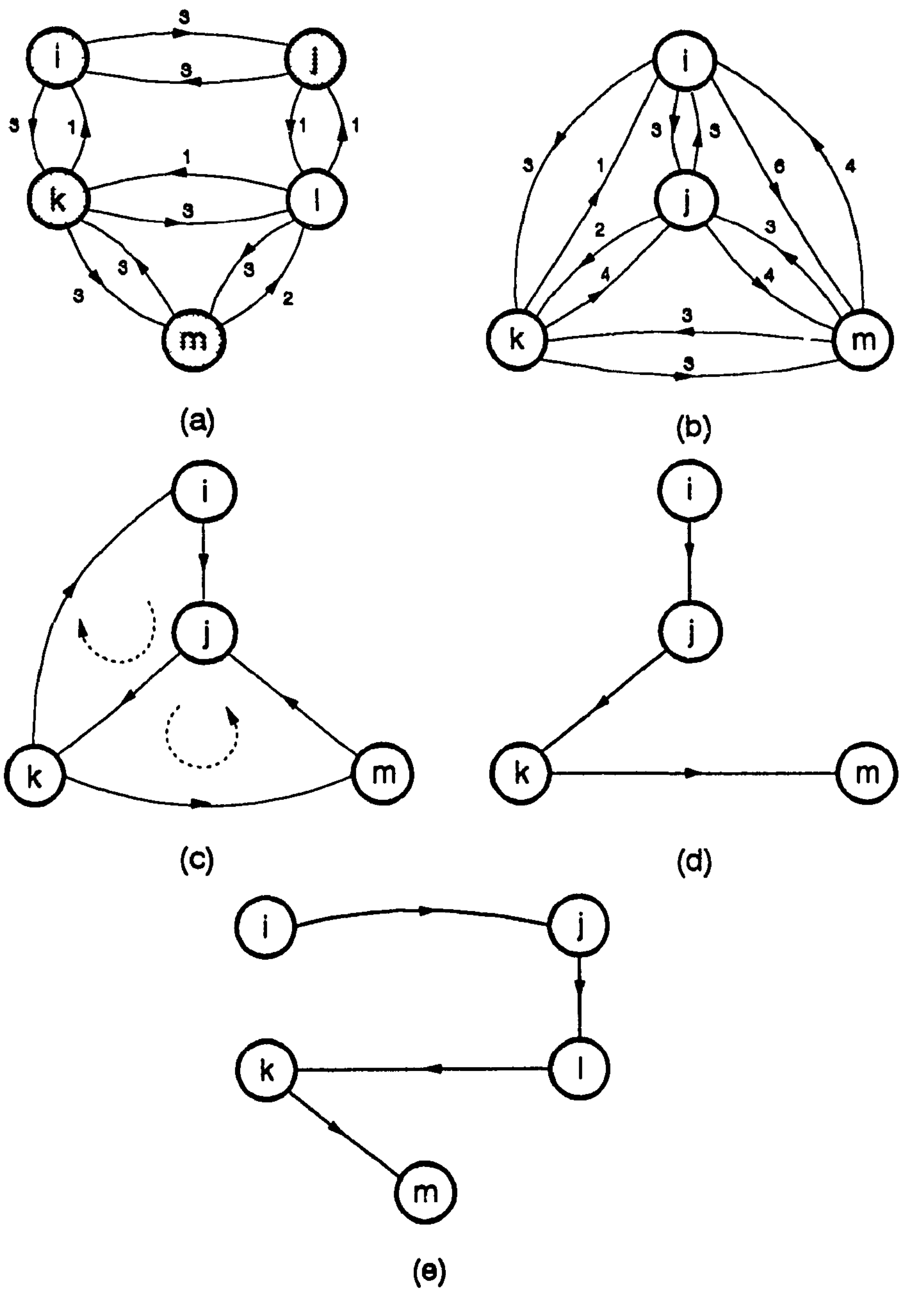


Fig. 5.1 The MDST heuristic. (a) The graph (b) The closure (c) The directed cycle (c) The MDST (e) The arborescence

calculated as

$$D_{ij} = \frac{\partial T_{ij}}{\partial f_{ij}} = \frac{c_{ij}}{(c_{ij} - f_{ij})^2} \tag{5.6}$$

and an approximation of a Steiner tree is computed by using the Minimum Directed Spanning Tree heuristic presented in [53]. An example of the MDST heuristic is demonstrated in Fig. 5.1, with i being the source and $\{j, k, m\}$ being the destination set D. First, determine the shortest Path $sp(i, j)$ for every i, $j \in \{i\} \cup D$ to construct a strongly connected directed graph (Fig. 5.1(b)). Second, subtract a constant from the cost of the edges incoming to each node in the graph, so that their minimum cost becomes zero. For example, the edges (j, i), (k, i) and (m, i), which are incoming to node i, have cost of 3, 1 and 4 respectively. The minimum value of the cost of these edges is 1. Subtracting 1 from the cost of these edges, the cost of (k, i) becomes zero. Third, remove all nonzero cost edges, the remaining zero cost edges form directed cycles (Fig. 5.1(c)). Next, starting with optimal arborescence rooted at i, remove edges in cycle incoming to i—(k, i); the first node on a directed path from i is j; then choose node j as a root for a smaller arborescence, remove edges in cycles incoming to j—(m, j); the other edges of the arborescence can be determined recursively in the same way, until no cycles exist (Fig. 5.1(d)). Finally, the edges in the arborescence are expanded to the edges in the original graph (Fig. 5.19(e)).

The algorithm for multicast routing is initialized with routing traffic over an arbitrary tree, and terminates when the routing variables and tree topology remain unchanged after a round of iterations.

5.3 NUMERICAL RESULTS

We next apply our heuristic on an empty network shown in Fig. 5.2. The network is a simplified version of the NSFNet backbone, with 12 nodes and 30 directed links. Link capacities are all equal to 100 data units/second. Since the

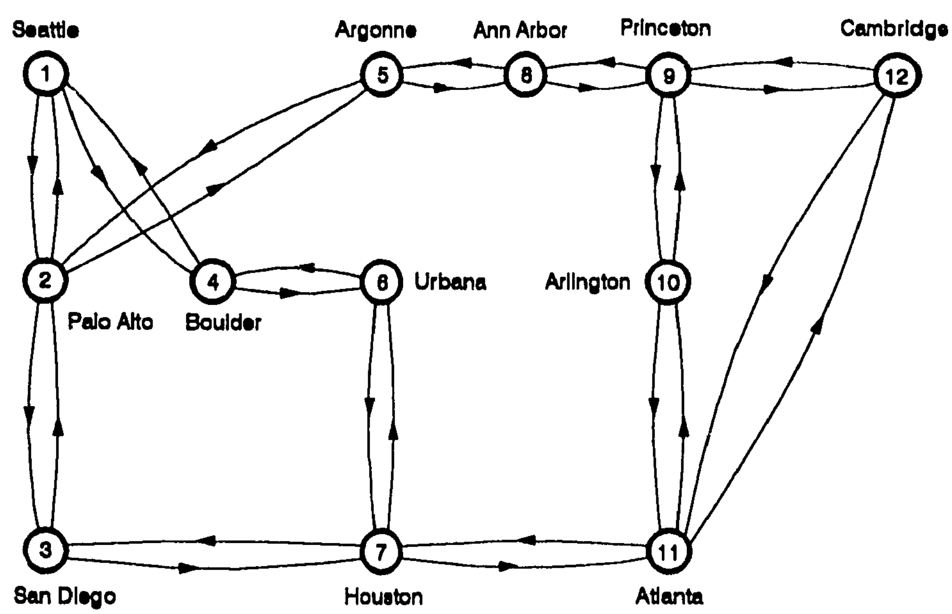


Fig. 5.2 The simplified NSFN backbone

capacity constraint was not taken into account in our formulation, the network was only exposed to a light traffic regime.

First, the algorithm has been examined on the network with 2 multicast streams:

Source	Destinations	Arrival rate (data units/second)
1	{10,11}	30
2	{10,12}	20

The initial flow allocation is depicted in Fig. 5.3, which gives an initial cost of 4.0358. We have observed that for each multicast, three trees arose as the algorithm progressed through its iterations. We chose $k = 2$ (the number of trees to be selected), and compared the result with the case that updating routing fractions over all arisen trees. The algorithm has reached to a valid solution in 8 and 12 iterations respectively. The final flow allocation of this example is shown in Fig. 5.4, and the routing fractions are listed in Table 5.1. As shown in Fig. 5.4, the algorithm performed well with $k = 2$, which gives a cost of 3.2622. It is close to the result yielded by considering all trees. According to our experiments for the network of Fig. 5.2 with various multicast streams, for each multicast choosing $k = \lceil$ (number of destinations + 1) / 2 $\rceil$ provides sufficiently good performance.

The run time of the algorithm is a function of number of links, number of multicasts, number of destinations in each multicast and also the number of trees considered in the algorithm. In Fig. 5.5, we plot the run time as a function of the number of destinations in the multicast, for 2, 4, 6 multicast streams in the network of Fig. 5.2 (the algorithm was implemented on a RISC 6000 workstation in C).

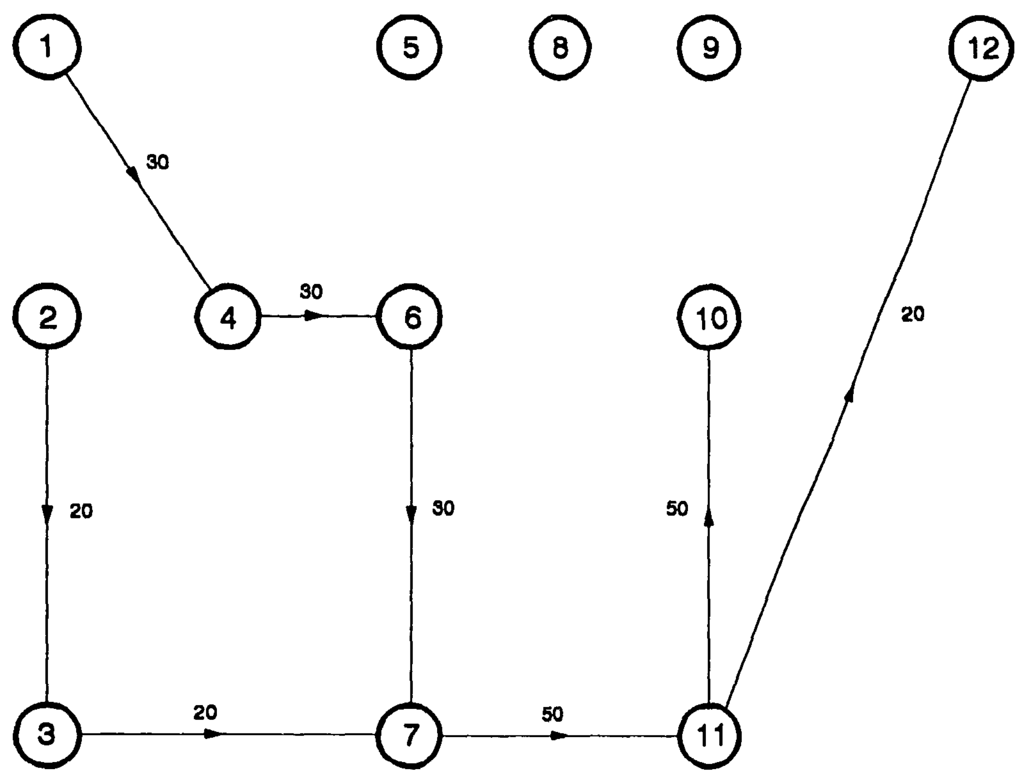
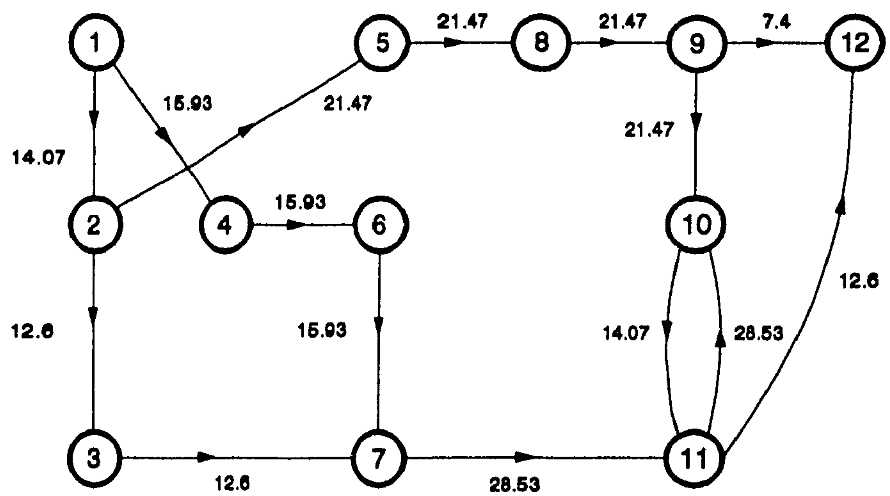
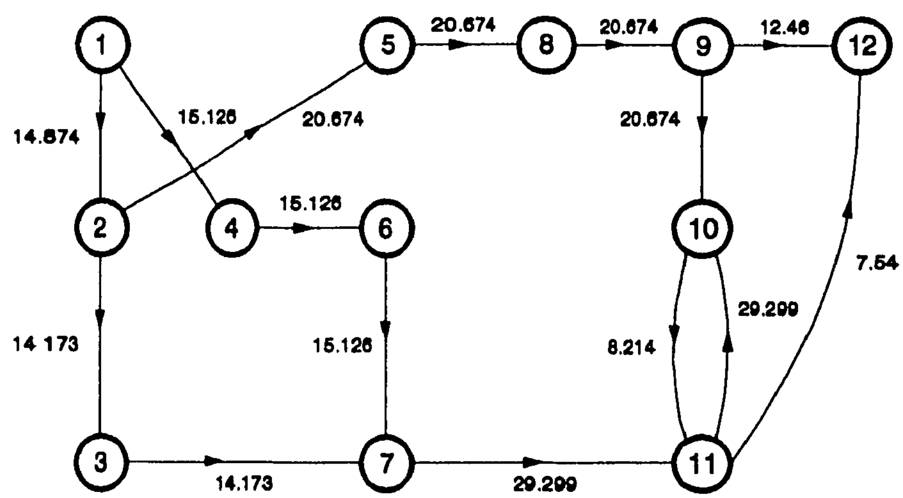


Fig. 5.3 The initial flow allocation

Cost = 4.0358



(a) k=2. Cost=3.2622



(b) All trees. Cost=3.224

Fig. 5.4 The final flow allocation

Table 5.1 Optimal routing fractions

Source	Destinations	Number of trees	Trees	Routing fraction
1	{10,11}	2	$1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 11 \rightarrow 10$	0.5310
			$1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$	0.4690
		all	$1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 11 \rightarrow 10$	0.5042
			$1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$	0.2738
			$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 11 \rightarrow 10$	0.2220
2	{10,12}	2	$2 \rightarrow 3 \rightarrow 7 \rightarrow 11 \rightarrow 10$ $\rightarrow 12$	0.6300
			$2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10$ $\rightarrow 12$	0.3700
		all	$2 \rightarrow 3 \rightarrow 7 \rightarrow 11 \rightarrow 10$ $\rightarrow 12$	0.3770
			$2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 12 \rightarrow 11 \rightarrow 10$	0.0000
			$2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10$ $\rightarrow 12$	0.6230

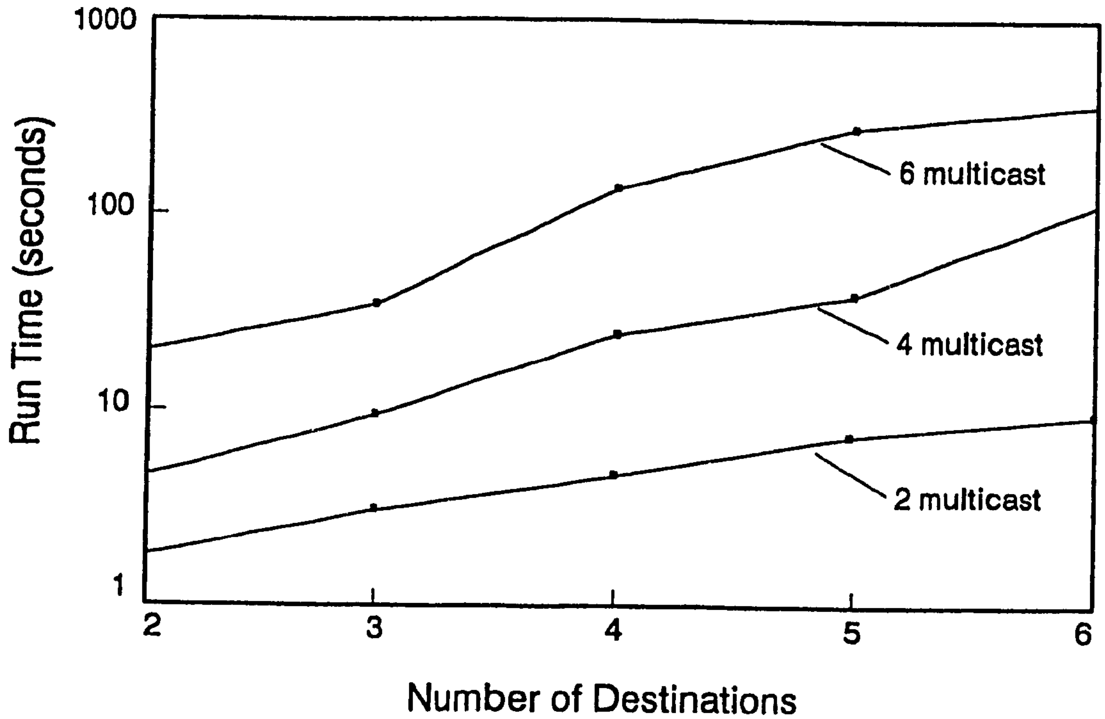


Fig. 5.5 Observed run times of the algorithm

The problem with the algorithm is that it might not find a route for multicast, when stream bandwidth requirements are a large fraction of the link bandwidth, even though one exists. It is necessary to take the bandwidth requirements into account and establish the convergence. This is a topic for further research.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

Routing and dimensioning in telecommunication networks are sophisticated network functions, which are made difficult as a result of the complicated interactions that take place among different parts of the network. The increased demand for multimedia services led to the requirement of developing efficient routing and dimensioning schemes for point-to-point and multipoint private networks capable of carrying multirate traffic. Our preliminary research has focused on these problems in cases where a third party carrier is available to handle overflow traffic on any link.

The first objective in this work is posing the optimization problems in circuit-switched networks as fixed-point problems using link and node flows. The fixed costs for leasing trunk capacities such as DS0, DS1, DS2 and DS3, and analytical models to measure call-blocking probabilities are used to construct a composite cost function that allocates a minimum-cost capacity over each link of a network as a function of its link load distribution, taking into consideration the cost of the overflow traffic. The optimization problems can then be solved using a flow-deviation algorithm. Since the capacity has been accommodated in a cost-effective fashion over each link of a network when the algorithms progress through their iterations, the dimensioning process is performed at the same stage with the routing process.

Assuming that the third-party carrier can offer the higher-rate circuit-switched capacity between all pairs of points, integrated point-to-point voice and video traffic are then incorporated into the optimization problem in the same way as handling single-service traffic. Conceptually, the formulation of this optimization problem allows for any number of classes of traffic with various rate requirements but only two are considered in the thesis for simplicity. Also, under the assumption that the third party needs only to provide point-to-point capacity for the network to provide multipoint services, a cost function can be specified and the multipoint optimization problem can be posed in a similar framework.

Employing a cost function that is the expression of the average number of packets (messages) in the data network, the algorithms for multipoint optimization problem in circuit-switched networks are applicable for optimally routing multicast packet-switched traffic.

Although the results of substantial experiments are promising, there is still a number of elements that remain to be investigated.

— Since hubbing between nodes is allowed, the order of iterations for optimal route calculation may be important.

— A simpler way to measure the blocking probability for each traffic type in multiple-service networks will be searched to obtain faster iterations.

— For practical multipoint connections, the bandwidth requirements for calling and called-parties are different. One example is a video-lecture, which can be established as a set of simultaneous multipoint calls that are logically related.

— If the bandwidth requirements are taken into account for multicast routing, the way of pruning of the links that do not have enough free bandwidth to support the multicast remains a problem.

— Theoretical research is needed regarding the convergence conditions and freedom from looping of the proposed algorithms.

— The algorithm are suitable for distributed computation.

Bibliography

[1] Bear, D., 1976, **Principles of Telecommunication-traffic Engineering,** *Peter Peregrinus Ltd.*

[2] Grandjean, C., 1967, "Call routing strategies in telecommunication networks", *Proc. 5th ITC*, pp. 3.2.B.4.1-9.

[3] Girard, A., 1990, **Routing and Dimensioning in Circuit-switched Networks**, *Addison-Wesley.*

[4] Ash, G. R., Cardwell, R. H. and Murray, R. P., 1981, "Design and optimization of networks with dynamic routing", *Bell System Technical Journal*, vol. 60, no. 8, pp. 1787-1820.

[5] Benes, V. E., 1966, "Programming and control problems arising from optimal routing in telephone networks", *Bell System Technical Journal*, vol. 45, no. 9, pp. 1373-1438.

[6] Heyman, D. P. and Sobel, M. J., 1984, **Stochastic Models in Operations Research, Vol. 2**, *New York: McGraw-Hill.*

[7] Krishnan, K. R., 1990, "Markov decision algorithms for dynamic routing", *IEEE Communications Magazine*, vol. 28, no. 10, pp. 66-69.

[8] Fratta, L., Gerla and Kleinrock, L. 1973, "The flow deviation method: an approach to store-and-forward communication network design", *Networks*, vol. 3, pp. 97-133.

[9] Gallager, R. G., 1977, "A minimum delay routing algorithm using distributed computation", *IEEE Trans. on Commun.*, vol. com-25, no. 1, pp. 73-85.

[10] Gavish, B. and Hantler, L., 1983, "An algorithm for optimal route selection in SNA networks", *IEEE Trans. on Commun.*, vol. com-31, no. 10, pp. 1154-1159.

[11] Mason, L. G., 1985, "Equilibrium flows, routing patterns and algorithms for store-and-forward networks", *Large Scale Systems*, vol. 8, pp. 187-208.

[12] Caron, F., 1988, "Results of the Telecom. Canada high performance routing", *Proc. 12th ITC*, pp. 3.1.B.2.1-7.

[13] Ash, G. R., 1985, "Use of a trunk status for real-time DNHR", *Proc. 11th ITC*, pp. 3.3.C.1.1-6.

[14] Gauthier, P. and Chemouil, P., 1987, "A system for testing adaptive traffic routing in France", *Proc. Globecom'87*, pp. 11.1-5.

[15] Krishnan, K. R., 1989, "Network control with state-dependent routing", *ITC Spec. Sem., Adelaide, Australia*, pp. 301-306.

[16] Stacey, R. R. and Songhurst, D. J., 1987, "Dynamic alternative routing in the British Telecom Trunk Network", *Proc. ISS'87*, pp. 3.1-7.

[17] Mase, K. and Yamamoto, H., 1990, "Advanced network traffic control methods for network management", *IEEE Commun. Mag.*, vol. 28, no. 10, pp. 82-88.

[18] Mason, L. G. and Girard, A., 1982, "Control techniques and performance models for circuit switched networks", *Proc. Conf. on Decision and Control*, pp. 1374-1383.

[19] Kelly, F. P., 1988, "Routing in circuit-switched networks: optimization, shadow prices and decentralization", *Adv. Appl. Prob.*. vol. 20, pp. 112-144.

[20] Kelly, F. P., 1990, "Routing and capacity allocation in networks with trunk reservation", *Math. of Oper. Res.*, pp. 422-457.

[21] Dziong, Z., Pióro, M., Kröner, U. and Wickberg, T., 1988, "On adaptive call routing strategies in circuit-switched networks", *Proc. 12th ITC*, pp. 3.1.A.5.1-8.

[22] Ash, G. R., Chen, J. S., Frey, A. E. and Huang, B. D., 1991, "Real-time network routing in a dynamic class of service network", *Proc. 13th ITC*, pp. 2.1.B.1.1-6.

[23] Kolarov, A. and Hui, J., 1994, "Least cost routing in multiple-service networks", *IEEE INFOCOM'94*, vol. 3, pp. 11c.1.1-8.

[24] Bertsekas, D. and Gallager, R., 1987, **Data Networks**, *Prentice-Hall*.

[25] Bharath-kumar, K. and Jaffe, J. M., 1983, "Routing to multiple destinations in computer networks", *IEEE Trans on Commun.*, vol. com-31, no. 3, pp. 343-351.

[26] Deering, S. E. and Cheriton, D. R., 1990, "Multicast routing in datagram internetworks and extended LANs", *ACM Trans. on Computer Sys.*, vol. 8, no. 2, pp. 85-110.

[27] Moy, J., 1993, "Multicast extensions to OSPF", *IETF Internet Draft.*

[28] Kou, L. Markowsky, G. and Berman, L., 1981, "A fast algorithm for steiner trees", *Acta Informatica 15*, pp. 141-145.

[29] Kompella, V. P., Pasquale, J. C. and Polyzos, G. C., 1993, "Multicast routing for multimedia communication", *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 286-292.

[30] Noronha Jr., G. A. and Tobagi, F. A., 1994, "Optimum routing of multicast streams", *IEEE INFOCOM'94*, vol. 2, pp. 7a.4.1-9.

[31] Kashper, A. N., 1987, "Bandwidth allocation and network dimensioning for International Multiservice Networks", *Proc. 5th ITC Seminar*, pp. 293-298.

[32] Schwartz, M., 1986, **Telecommunications Networks: Protocols, Modeling and Analysis**, *MA: Addison-Wesley.*

[33] Mason, L. G and Larocque, M., 1987, "Dimensioning rearrangable transmission network with survivability constraints", *INRS-Telecommunications, Tech. Rep.*

[34] Yoshida, M. and Okazaki, H., 1988, "New planning architecture for reliable and cost-effective network design", *Proc. 12th ITC*, pp. 4.1.A.1.1-5.

[35] 1980, "Special issue on Telecommunications network design and planning", *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 8.

[36] Key, P. B., 1988, "Implied cost methodology and software tools for a fully connected network DAR and trunk reservation", *Brit. Telecom. Tech. J.*, vol. 6, no. 3, pp. 24-35.

[37] Shinohara, M. and Kimura, G., 1988, "A dimensioning method for hierarchical circuit-switched networks", *Rev. Elec. Commun. Lab.*, vol. 36, no. 1, pp. 35-39.

[38] Herzog, U., 1970, "Calculation of two-way trunk arrangements with different types of traffic input", *Proc. 6th ITC*, pp. 217-223.

[39] Wilkinson, R. I., 1956, "Theories of toll traffic engineering in the U.S.A.", *Bell System Technical Journal*, vol. 35, no. 2, pp. 421-514.

[40] Fredericks, A. A., 1980, "Congestion in blocking systems—A simple approximation technique", *Bell System Technical Journal*, vol. 59, no. 6, pp. 805-827.

[41] Akimaru, H. and Nishimura, T., 1963, "The derivatives of Erlang's B formula", *Rev. Elec. Commun. Lab.*, vol. 11, no. 9-10, pp. 1215-1228.

[42] Jagerman, D. L., 1984, "Methods in traffic calculations", *Bell System Technical Journal*, vol. 63, no. 7, pp. 1283-1309.

[43] Gimpelson, L. A., 1965, "Analysis of mixture of wide and narrow-band traffic", *IEEE Trans. Commun. Technol.*, vol. com-13, no. 9, pp. 1411-1440.

[44] Enomoto, O. and Miyamoto, H., 1976, "An analysis of mixtures of multiple bandwidth traffic on time division switching networks", *NEC Res. and Devel.*, no. 41, pp. 73-81.

[45] Fredrikson, G. F. W., 1974, "Analysis of channel utilization in traffic concentrators", *IEEE Trans. on Commun.*, vol. com-22, vol. 8, pp. 1122-1129.

[46] Frenkel, G., 1974, "The grade of service in multiple access satellite communication systems with demand assignment", *IEEE Trans. on Commun.*, vol. com-22, vol. 10, pp. 1511-1518.

[47] Kaufman, J. S., 1981, "Blocking in a shared resource environment", *IEEE Trans. on Commun.*, vol. com-29, no. 10, pp. 1474-1481.

[48] Hwang, F. K. and Richards, D. S., 1992, "Steiner tree problems", *Networks*, vol. 22, no. 1, pp. 55-89.

[49] Horowitz, E., 1982, **Fundamentals of data structures**, *Computer Science Press, Inc.*

[50] Kruskal, J., 1956, "On the shortest spanning tree for a graph and travelling salesman problem", *Proc. American Math. Society*, vol. 7, pp. 48-50.

[51] Kleinrock, L., 1964, **Communication networks: stochastic message flow and delay**, *New York: McGraw - Hill.*

[52] Hayes, J. F., 1984, **Modeling and analysis of computer communications networks**, *New York: Plenum Publishing Co.*

[53] Humblet, P. A., 1983, "A distributed algorithm for minimum weight directed spanning tree", *IEEE Trans. on Commun.*, vol. com-31, no. 6, pp. 756-762.