A VIRTUAL PATH ROUTING ALGORITHM FOR ATM NETWORKS BASED ON THE EQUIVALENT BANDWIDTH CONCEPT

by

Kaan Bür

B.S. in Control and Computer Engineering, İstanbul Technical University, 1995

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Engineering

Bogazici University Library
39001100121139

Boğaziçi University

1998

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Cem Ersoy, not only for his supervision in this M.S. study, but also for his support and encouragement during the past three years of my graduate education.

To January 26, 1997...

ABSTRACT

The coexistence of a wide range of services with different quality of service (QoS) requirements in today's networks makes the efficient use of resources a major issue. It is desirable to improve network efficiency by adaptively assigning resources to services that have different bandwidth demands. Implementing Broadband Integrated Services Digital Networks (B-ISDN) therefore requires a network control scheme that can absorb unexpected traffic fluctuations. Asynchronous Transfer Mode (ATM) technology provides this flexibility by virtualizing network resources through the use of the virtual path (VP) concept. The traffic demand of new services in a B-ISDN environment may be highly bursty and difficult to predict. The implementation of the equivalent bandwidth concept provides an efficient method to estimate capacity requirements. The concept also defines a unified connection metric to be used for network management, routing and optimization. In this study, a method for designing a VP-based ATM network is proposed. The developed heuristic design algorithm uses the equivalent bandwidth concept to compute the capacity requirements of the connection requests. This way, the desired QoS defined by the cell loss probability is guaranteed. The algorithm applies VP routing and separation techniques to minimize the maximum link utilization under processing delay constraints. The quality of the solutions achieved by the heuristic design algorithm is compared to several competitors under varying network topologies and traffic conditions. The observations on the algorithm performance show that the developed method is able to facilitate an efficient use of network resources through the introduction of VPs.

ÖZET

Günümüz bilgisayar ağlarında her biri ayrı servis kalitesi (QoS - quality of service) gerektiren birçok değişik hizmetin bir arada varolabilmesi, kaynakların etkili kullanımını temel bir konu durumuna getirir. Kaynakları, değişik bant genişliği gereksinimleri olan servislere uyum sağlayacak biçimde atayarak ağ etkinliğini artırmak aranan bir özelliktir. Bu yüzden, Genişbantlı Tümleşik Servisler Sayısal Ağları'nın (B-ISDN - Broadband Integrated Services Digital Networks) gerçeklenmesi trafikteki beklenmedik değişiklikleri karşılayabilecek bir denetim düzeneği gerektirir. Eşansız Aktarım Kipi (ATM - Asynchronous Transfer Mode) teknolojisi, sanal yol (VP - virtual path) kavramıyla ağ kaynaklarını sanallaştırarak bu esnekliği sağlar. B-ISDN ortamındaki yeni hizmetlerin trafik gereksinimleri yüksek oranda patlamalı, öngörülmesi zor olabilir. Eşdeğer bant genişliği kavramının uygulanması, kapasite gereksinimlerinin kestirilebilmesi için etkili bir yöntem sağlar. Bu kavram aynı zamanda ağ yönetimi, yönlendirme, en iyi biçimde kullanım için ortak bir bağlantı ölçütü tanımlar. Bu çalışmada, VP temelli bir ATM ağ tasarım yöntemi önerilir. Akıl yürütme yoluyla geliştirilen tasarım algoritması, bağlantı isteklerinin kapasite gereksinimlerini bulmak için eşdeğer bant genişliği kavramından yararlanır. Bu yolla, hücre kayıp olasılığı ile tanımlanan QoS yerine getirilir. Algoritma, işlem gecikmesi kısıtları altında en yüksek fiziksel hat kullanım oranını en aza indirmek üzere, VP yönlendirme, VP ayrıştırma teknikleri uygular. Tasarım algoritmasının elde ettiği çözümlerin kalitesi, değişken ağ topolojileri ile trafik koşulları altında bir dizi sonuçla karşılaştırılır. Algoritmanın başarımı üzerinde yapılan gözlemler, geliştirilen yöntemin VP'ler tanımlayarak ağ kaynaklarının etkin biçimde kullanımını kolaylaştırdığını gösterir.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b_k	Mean of the burst period of the k^{th} connection on a VP
C	Set of all VCs
C_{ij}	Set of all VCs passing through l_{ij}
c_{ij}	VC with source-destination pair i and j, $(i,j \in V)$
c'_k	Equivalent capacity of the k^{th} connection in isolation on a VP
C'	Equivalent capacity of n connections multiplexed on a VP
$C'(F)$	Fluid-flow approximation to C'
$C'(S)$	Stationary approximation to C'
d_{ij}	Physical capacity of l_{ij}
E	Set of all network links
h	Maximum number of VP hops
L	Connection metric vector of a VP
l_{ij}	Link connecting adjacent nodes i and j, $(i,j \in V)$, $(l_{ij} \in E)$
m	Mean aggregate bit rate of n connections multiplexed on a VP
m_k	Mean of the peak rate of the k^{th} connection on a VP
n	Number of multiplexed connections on a VP
P	Set of all VPs
P_{ij}	Set of all VPs passing through l_{ij}
p_{ij}	VP with source-destination pair i and j, $(i,j \in V)$
$R_{peak,k}$	Peak rate of the k^{th} connection on a VP
t_{ij}	Traffic demand of a VC c_{ij}
u_{ij}	Link utilization of l_{ij}
V	Set of all network nodes
x	Buffer size of the sources
β	Connection characteristics parameter
σ^2	Variance of the aggregate bit rate of n connections multiplexed on a VP
σ_k^2	Variance of the peak rate of the k^{th} connection on a VP
ε	Maximum cell loss probability (buffer overflow probability)
ρ_k	Source utilization (fraction of active time of the k^{th} connection on a VP)

1. INTRODUCTION

A substantial amount of research effort in communication network engineering has been spent on service integration during recent years. The future telecommunications network should provide basic communications capabilities in an application-independent fashion. Various new information services, such as video on demand (VoD), video teleconferencing, etc., have to be handled in the most efficient and economical way possible while traditional ones like data transfer services are maintained. A key characteristic of these services is that they require quality of service (QoS) guarantees [1]. The information bit rates transported in the network will range from kilobits to gigabits per second, and the statistical nature of each type of service will be different. This wide range of bit rates and different statistical natures have to be handled efficiently by the network. The network should also be able to evolve to meet unknown future demands for increased flexibility, capacity, and reliability [2].

Advances in high-speed multiplexing, switching and optical transmission systems have caused a great deal of interest in Broadband Integrated Services Digital Networks (B-ISDN), which support a wide range of applications with different QoS requirements in a flexible and cost-effective manner. Great efforts are being made to develop a flexible and reliable B-ISDN capable of transmitting signals from a diverse range of media. The network capabilities include support for the following issues.

(A) The network serves as a common carrier of both interactive and distributive communications, which may include audio, video and data.

(B) Switched services work at broadband rates of up to about 150 Mbps, with the possibility of raising this limit in the future.

(C) While continuous services like audio and video may need guaranteed bandwidth to meet performance requirements (low information loss and delay), bursty data services may be provided more cost-effectively with a lower grade of service (GoS), using resources that can be shared statistically.

(D) Some applications, including most of the constant bit rate (CBR) communications, are best served by connection-oriented services. Connection-oriented services, including

real-time audio and video communications, are characterized by separation of the procedures for connection establishment and the end-to-end transfer of user information. Connection establishment procedures, which must precede information transfer, determine a route and set up a path between users. Connectionless services, including mail and other data-oriented communications, may not warrant separate connection establishment and information transfer phases, with information about route and user carried in the same message.

(E) Some services require transparent transport of digital bit streams through the network. Others may take advantage of code compression capabilities used to conserve internal bandwidth or code conversion capabilities provided to overcome terminal incompatibilities.

(F) Some services require a single point-to-point connection, either unidirectional or bi-directional, between two end points. Other "multimedia" applications benefit from parallel connections between end points, connections among multiple users, and even combinations of multiple-connection, multi-user calls. Some multi-user services may involve digital signal processing.

The goal of B-ISDN is to define a user interface and network that meets these varied requirements. An objective of the network is to be able to accommodate volatile changes in service mixes, both at the level of individual interface and the system as a whole. High-capacity and high-performance fiber-based transmission facilities are generally assumed to be required to support this environment. In addition, the network must have flexible switching capabilities and special-purpose service modules.

ITU-T Study Group XVIII, who are responsible for the international standards on B-ISDN, generically calls the switching and multiplexing aspects of B-ISDN the "transfer modes". The transfer mode chosen by the ITU-T Study Group XVIII Task Force on ISDN Broadband Aspects as the basis of B-ISDN is called the Asynchronous Transfer Mode (ATM). ATM is a high-bandwidth, low-delay, packet-like switching and multiplexing technique which provides the required flexibility for supporting heterogeneous services in a B-ISDN environment [3]. ATM systems have become possible through technological innovations such as fiber-optics, VLSI, and microprocessors. In ATM, information is

organized into short fixed-length entities, called cells, which are provided with short identification headers. Cells are transported to their destinations according to their headers in a way similar to packet switching. This information transportation scheme enables the network to handle efficiently a wide range of information bit rates and various types of traffic with different statistical natures [4]. ATM can economically provide integrated transport capability at virtually any bit rate. Multiplexing and transport aspects of ATM are expected to have a great impact on the network architecture because hierarchical channel structures and resulting complicated frames can be eliminated. Thus, ATM ensures that the network has maximum flexibility, which will enable graceful network evolution against unknown user demands and technological advances [5].

Very large capacity fiber-optic transmission technologies have significantly reduced the transmission cost portion of the total network cost. The trend toward reduced transmission cost is expected to continue in the future, so the node cost will get relatively high. This leads to the conclusion that the total network cost would be reduced most effectively by node cost reduction, which could be achieved by simplifying both node functions and transport network architecture. Simplification of network architecture and node processing is the key to developing a cost-effective, flexible network. This will be possible by implementing the virtual path (VP) concept. Introduction of the VP concept into an ATM network allows management of virtual circuits (VC) by grouping them into bundles. Consequently, VCs can be transported, processed, and managed in bundles, which permits significant advantages, including great reduction in node costs and simplification of both the transport network architecture and required operation, administration and maintenance (OAM) functions. However, such a simplification may slightly decrease the efficiency of utilizing transmission line capacity [2].

The VP concept has been proposed to realize a B-ISDN that takes maximum advantage of ATM. It is an important aspect of current ITU-T recommendations on B-ISDN. The concept is now recognized as providing a powerful transport mechanism with ATM networks. The fundamental advantage of this concept is that it allows the grouping of individual connections, also known as VCs, sharing common paths through the network to be handled and switched together as a single unit. Network management actions can then

be applied to a small number of groups of connections instead of a large number of individual connections, resulting in smaller total processing requirements, faster processing per VC, and in general, a significantly better use of network resources. More than 90 per cent of processing time can be saved when VCs are routed on VPs rather than processed individually [6].

The impact of VPs takes place in three areas, namely call acceptance and setup, cell routing, and adaptability to changes. Employing VPs reduces the processing and delay associated with the call acceptance control (CAC) functions. Thus, VPs play an important role in connection admission control in B-ISDN [7]. The VP approach will not only reduce node processing required at call establishment but also simplify network architecture, thus increasing network operability and reliability. By reserving capacity on a VP connection in anticipation of later call arrivals, new VC connections can be established by executing simple control functions at the endpoints of the VP connection. No call processing is required at transit nodes in this case. As a result, the VP concept will realize cost-effective networks with enhanced performance [8].

An ATM network is essentially constructed with nodes, physical links such as transmission lines, and terminals. A VC is defined by creating a connection between two terminals (users) which are exchanging information. A route is assigned to each VC in the network. Cells are transferred along the route assigned to the VC to which they belong. In this network, several node functions are needed. First, each node needs to recognize the outgoing link or terminal to which incoming cells should be sent. They must also determine the route for the VC upon call request and rewrite routing tables to set up a route. If sufficient bandwidth can not be reserved in the network for the requested connection, the call should be rejected. This way, specified transmission qualities such as cell delay and cell loss are assured for all connected VCs. Thus, the decision to refuse or accept a call at call setup is another necessary node function.

The VP concept simplifies and eliminates some of these node functions. In particular, it reduces processing loads generated at call setup. VPs are logical information transport paths defined as follows.

(A) A VP is a logical direct link between two nodes and accommodates a number of VCs simultaneously.

(B) A predefined route is defined for each VP in the physical facilities network.

(C) Each VP has a bandwidth, in other words, "capacity" which defines the upper limit for the total VC bandwidth carried by it.

(D) VPs are switched on physical transmission links in a cell multiplexing manner.

By using VPs with the mentioned features, the functions needed to set up a path through the network have to be executed only once for all VCs using that path and the required node functions are effectively simplified. First, it becomes unnecessary to rewrite the routing table of the transit nodes at call setup. This is made possible by reserving an area termed Virtual Path Identifier (VPI) in each cell header. At the transit nodes through which the VP passes, an outgoing link for arrived cells can be recognized by comparing their VPIs with the routing table. The cell processing time is reduced compared to VC switching, which uses both VPI and Virtual Circuit Identifier (VCI) fields. Since the routing table is concerned with VPs and not associated with each cell, rewriting of the table is not necessary at call setup. Routing procedure at call setup is also eliminated at the transit nodes because this is done by selecting the most appropriate VP from the end nodes terminating the VP. Additionally, the transit nodes are free from the bandwidth allocation process at call setup. This can be carried out by comparing the bandwidth of the requested call to the unused bandwidth of the VP at the end nodes. Since VPs have guaranteed bandwidth along their path, these functions have to be performed only at the beginning node of a VP. By reserving capacity on VPs, VC connections can be established quickly and simply.

Thus, all call setup functions can be eliminated from the transit nodes in the ATM network based on the VP concept. This effectively reduces the processing loads of the transit nodes and leads to low cost node construction. This characteristic is valuable in constructing an economical ATM network since transmission costs have been remarkably reduced because of the development of high capacity optical transmission systems. Other advantages of using VPs include logical service separation on network service access and adaptability to varying traffic and network failures through dynamic resource management.

It is possible to implement priority control by segregating traffic with different QoS, where each VP is considered a logical link for a certain service. Thus, VP subnetworks for different services can be built within an ATM network. Each VP is assigned a number of physical links and an effective capacity (in terms of the maximum number of VCs allowed) to assure QoS requirements. Several VPs may be multiplexed on the same physical link. Varying traffic conditions and network failures can be managed by adjusting the VP network to accommodate the changing conditions and to maintain network performance.

On the other hand, the disadvantage of using VPs is that the concept is based on reserving capacity in anticipation of expected traffic, which means decreased capacity sharing, i.e., decreased utilization of available transmission capacity. In other words, a VP cannot exploit redundant capacity on another VP, and capacity segmentation occurs. Consequently, the network throughput decreases as the total call blocking rate and the network transmission costs increase.

Dynamic bandwidth control is an efficient method of flexibly reassigning individual VP bandwidths. When the connections on one VP increase, remaining link capacity is assigned to the busy VP. This is made possible by the statistical sharing of transmission facilities among VPs. Thus, transmission efficiency is improved because each VP in the link is well utilized. Although this control may increase the processing load, the advantage in reduced node processing is expected to be maintained by changing the bandwidth less frequently than call setup and clearance. Since the VP bandwidth can be varied by merely modifying the bandwidth data stored in the control processor of the end nodes and not by accessing the switches, the transit nodes and all the switches do not need to be accessed for bandwidth changes. The VP bandwidth can be set at the smallest value needed for containing the VCs [4].

The purpose of this study is to develop a method of VP routing and bandwidth allocation in ATM networks. The proposed method applies dynamic capacity control in order to meet QoS requirements such as limited delay and bounded cell loss probability. It also distributes the network traffic in such a way that: (a) the effect of link failures is kept as small as possible; (b) link saturations are rare; (c) the network robustness is increased.

As a result, the method can facilitate an efficient use of the network resources. This study is organized as follows: Chapter 2 contains a review of the previous work, where different approaches, goals and strategies related to the problem are summarized to show how the VP design problem has been attacked in the literature. Chapter 3 introduces the objective, the constraints and the network model. The formulation of the problem and the definitions of some terms related to this study are covered in this chapter. In Chapter 4, the VP routing algorithm, its phases and key functions are described. Chapter 5 presents computational experiments, results, and comments on the algorithm performance. Finally, a summary, conclusions and subjects for further work are given in Chapter 6.

2. REVIEW OF THE RELATED WORK

All VP routing and capacity allocation algorithms fall into two major categories, namely "synchronous" and "asynchronous". Synchronous algorithms update the VP capacity in real-time based on the observed demand for call establishment. In this context, the bandwidth of VPs can be expanded or contracted upon call arrival or departure. On the other hand, asynchronous algorithms maintain a fixed VP distribution for a time period referred to as the update interval. These algorithms are called asynchronous, because the modifications of VPs and their capacities are not driven by the call arrival or departure process associated with each VP. The VP distribution policy is computed at the beginning of the period and remains fixed during that period.

Asynchronous algorithms can be further divided into two types, which are "centralized" and "distributed". Centralized algorithms run in one location and require the collection of up-to-date information from all network nodes, while distributed algorithms run in every network switch and information is passed along between neighboring nodes. Centralized implementations employ a central route server that is "all knowing" and stores the topology and resource profile of the network. Switches store local copies of the database for local switching and forward requests that are unknown to the local database to the server. Updates to the topology are first installed on the server and then distributed via routing table downloads. Changes in the QoS capacity of a link due to the increase in traffic, as well as local link outages, must be first propagated to the route server and then sent to all the switches. Distributed routing algorithms can be more complex than centralized ones, and in terms of resource usage, they offer a suboptimal solution to the problem. However, their implementations are far more scalable.

To identify the key characteristics of VP routing and capacity allocation algorithms, some performance issues affected by the design of a VP network have to be considered. A good VP layout is characterized by achieving a good performance trade-off among them.

(A) Low VC setup complexity is important as it substantially reduces various overheads of connection management. The setup complexity is proportional to the number of nodes in the VC, in which the VCI is examined, since in these nodes the intervention of software is needed (to change the VC routing tables and allocate bandwidth). For this reason, the number of VPs used for routing a VC (termed VP hop count) should be small.

(B) The chosen route for a VC must also be short in terms of the number of physical links it uses, or better in terms of propagation delay, to efficiently utilize the communication network.

(C) The number of occupied entries in the VP routing tables (termed the load on the table) implied by the layout should be low enough at any location in the network.

(D) The resulting layout must overcome link disconnections with a low recovery overhead. This is achieved by reducing the number of VPs that share any link, so that if a link is disconnected, the number of VPs that need to be rerouted, in order to by-pass the faulty link, will be small [9].

Based on these briefly mentioned issues, algorithms can also be categorized by the objective (cost) function employed in the process of VP routing and capacity assignment. If the objective of the VP distribution problem is to achieve a satisfactory network throughput and guarantee QoS at the call level, it is logical to include the call blocking rates in the cost function. The call blocking rate is determined as the probability that a call for a new connection is rejected because not enough resources can be found to guarantee QoS of all existing connections and the new connection. In a network design where they are not considered, the call blocking rates might be acceptable only for some source-destination pairs and violating the constraints for others, even if the network throughput is maximized. By incorporating the total rejected bandwidth demand in the cost function, the total carried capacity in the network is maximized. This has some advantages because the call blocking rates, which are non-linear functions of the VP capacities, do not participate in the cost function, and the optimization problem becomes more tractable.

2.1. Design Goals

Usually, the network topology is modeled by a directed graph $G = (V,E)$, where V is the set of vertices (nodes) and E is the set of edges (links). It is often assumed that G is strongly connected, i.e., there is a directed path from any node to any other. A VP is defined by a starting vertex and an ending vertex, a directed route between these vertices, and a capacity assigned to this connection. The starting and ending vertices of a VP are called VP terminators. The capacity, also referred to as demand or bandwidth, expresses the fact that a given portion of the transmission capacity is reserved for the VP to serve the traffic demand brought about it. Sometimes it is assumed that the capacities are normalized to their maximum, that is, capacities are measured in relative units that guarantee that their sum is one and do not exceed the total link capacity. This way, capacity can also be regarded as the fraction of bandwidth used by the VP on a unit capacity link [6, 10].

2.1.1. Maximum Virtual Path Blocking Rate

A possible objective is minimizing the maximum VP blocking rate with the typical constraint of traffic loss along VPs. All traffic for each source-destination pair should be serviced, and the aggregate flow on the links should not exceed the effective link capacity to satisfy QoS requirements. If there are delay limitations, another constraint should be included to keep the average number of hops for a VP below a given constant. In general, the maximum VP blocking rate does not increase when the degree of channel sharing among source-destination pairs increases [11]. A capacity allocation algorithm is applied to reallocate VP capacities after each move in the VP assignment algorithm. This algorithm tries to improve the blocking probabilities of the worst-blocked traffic streams by adding capacity to VPs used by these traffic streams until no more improvement is possible in the blocking probabilities [12]. In another study, Shioda employs a cost function based on the VP blocking probability for selecting the VP that will get a capacity increase [13].

2.1.2. Delay Requirements

Among all the QoS requirements imposed by potential ATM network users, that of guaranteeing the timely delivery of messages has drawn considerable attention, especially in the domains of voice/video data transmission over a data network, and message communication in an embedded real-time system. The QoS guarantee on the delay bound and the delay variance is not possible without a network protocol/scheme that supports the timely and predictable delivery of messages. Given an ATM network, there are two issues to be considered. These are the issue of laying out VPs, and the issue of selecting VC routes along which sufficient resources are reserved to meet the user-specified end-to-end delay requirements. The objective in this case is to ensure that each message of the VC will be delivered to its destination node with a time period no longer than the user-specified end-to-end delay bound after its arrival, thus satisfying the timing requirements for that VC, while not jeopardizing the QoS guarantees to other existing VCs. Since VCs are established and terminated dynamically with time, and since the number of all possible routes between a given pair of nodes may be very large, it is very difficult, if not impossible, to find an optimal VC route that can meet all these requirements. Therefore, the following two principles are used to find a VC route to be established.

(A) The selected VC route must survive the admission test. Each VP terminator on a candidate route conducts a test on the VP and calculates the minimum worst-case traversal time which a message of the VC will experience when it traverses the VP. A VC route is said to survive the test if there is sufficient bandwidth on each VP on the route so that the sum of the worst-case traversal times on all the VPs on the route does not exceed the specified end-to-end delay.

(B) The selected VC route is the shortest VC route among all routes that survive the admission test. In other words, the number of VPs along which the VC is routed is minimum among all possible VC routes that survive the admission test. This way, the setup time for the call setup request will be reduced by selecting a VC route with a minimum number of VPs, and the network call blocking probability for future VCs may be reduced by using as few network resources (VPs and their bandwidths) as possible [14]. This implementation is favorable when the call setup time is considered

a critical performance measure and the network has abundant capacity so that the achievable total call blocking rate is within an acceptable range [11].

2.1.3. Network Construction Cost

In an attempt to design the VP-based ATM network, it is required to consider a network topology and traffic pattern generated from users for minimizing a network construction cost (or, as an alternative, processing/control costs at nodes and transmission costs between nodes [12]) while satisfying QoS requirements such as cell/call loss probabilities and cell delay times. To achieve this goal, a minimum bandwidth required to transfer a given amount of traffic is first obtained by utilizing an equivalent bandwidth method [15]. After all the routes of VPs are temporarily established by means of the shortest paths, the network cost is minimized through the alteration of VP routes, and the separation of a single VP into several VPs. The effective bandwidth of each VP and the route of each VP are variables to be jointly determined to minimize the network construction cost. The link cost is formulated as a linear function of the number of physical lines, instead of the total bandwidths of VPs passing through the link. In short, the network cost is minimized by eliminating unnecessary links. More precisely, first the maximum number of VCs which should be admitted simultaneously for each VP to satisfy the constraint of call loss probability at call level is determined. Then, for the given number of VCs, the minimum required bandwidth of the VP is found by utilizing an equivalent bandwidth method under the constraint of cell loss probability at cell level. Since the equivalent bandwidth method does not take into account the delay times, however, the number of hops of every VP is restricted in determining its route [16, 17].

2.1.4. Total Network Revenue

Another possible objective function is the total network revenue summed up over the whole ATM network. It is assumed that each accepted connection will generate revenue at a given rate depending on the traffic type and route, and the total network revenue can then be seen as a weighted sum of carried traffic values [18]. The VP distribution problem can be seen as a network control problem with the following formulation: Given the

network topology, the capacities of network links, and the matrix of offered network load, calculate the routes and capacities of VPs in the network such that: (a) The sum of VP capacities on each link does not exceed its capacity. (b) The call blocking rate of each source-destination pair is below a predefined upper bound referred to as the source-destination blocking constraint. (c) The network revenue is maximized.

The algorithm tries to maximize the network revenue (considered as the weighted sum of the throughputs for each traffic class and source-destination pair multiplied by the respective call holding times), while satisfying the node processing and source-destination blocking constraints [19].

2.1.5. Maximum Link Utilization

Given the network topology, and the demand for VP capacity between specified pairs of nodes (VP terminators), one can choose VP routes connecting the terminators, such that the maximum link utilization is minimized. The load, or congestion of a link is defined as the summed capacity of VPs traversing the link. The system of VPs is optimal if the maximum link utilization is the smallest possible. The motivation behind this objective is ATM realization in bandwidth restricted systems. A viable VP system depends on the value of the maximum link utilization that can be achieved, making the chosen objective a primary design consideration.

The objective function can be changed to represent the ratio of the link utilization over its expected value in order to make the problem more feasible. The insight behind the new objective function is that the expected value of the load is independent of the actual value. Consequently, the expected value is a constant in the objective function, depending only on the probability distribution, but not on the actual VP route. Therefore, a choice of a VP that keeps the relative congestion small suggests itself as a reasonable approximation for keeping small the congestion itself. From the networking point of view one can argue that if link capacities are designed to serve the average demand, represented by the expected value, then a solution with a small maximum link overload ratio will decrease link utilization. The actual route of a VP is selected randomly from the set of all possible

shortest paths to make a more balanced load distribution and resource usage possible. In this context, each link is assigned a weight, and the shortest path is defined according to a function of the weights of the links involved [1, 6, 10].

2.2. Virtual Path Network Management Strategies

The resource management activities employed to facilitate VPs can be categorized by the time period on which they operate. Short term strategies are for dynamically making minor or incremental changes to the VP topology or capacity assignments. Medium term strategies are for making more widespread modifications to the VP overlay network. This would be appropriate for traffic changes based on time of day and for recovering from network failures. Finally, long term strategies may be employed to design a general VP overlay network, to be used at startup or after major network upgrades.

Short term strategies are referred to as successive modifications and are seen as adapting to small variations in the network traffic and minor failures. To ensure responsiveness, it has been suggested that the algorithms used should obtain results quickly, even at the expense of optimality, and that the changes should be minor in order to speed up the implementation phase. Furthermore, if modification decisions are made in a distributed manner, with local information and local objectives, changes can be made more quickly.

Medium term strategies are called global modifications. These strategies are for making network-wide changes to the topology and capacity assignments. They are seen as reacting to large fluctuations in network traffic. It is assumed that they occur much less frequently than successive modifications. Problems formulated in terms of global modifications typically attempt to optimize some network-wide control objective, thus involving a more complex, time-consuming algorithm. Changes to the VP overlay network are also much more widespread, thus increasing the time required to implement the changes. Due to the longer term and amount of information required for a global

modification, centralized control is considered a more attractive option than distributed control.

Long term planning strategies may be employed to design the VP overlay network. These strategies are likely based on knowledge of the physical network and possibly estimates of traffic patterns. The nature of long term planning imposes few constraints on the algorithm complexity, since solutions need not be found quickly [20].

Under the broad classification of these strategies, one can identify the following four basic VP management activities that are discussed in the literature.

(A) Capacity reallocation redistributes capacity on a fixed VP topology.
(B) Topology reconfiguration establishes and tears down VPs within an existing VP topology.
(C) Global configuration consists of both global capacity reallocation and global topology reconfiguration. This activity potentially affects all VPs in the network.
(D) Long-term planning derives a static or general set of VPs and initial or minimum capacity assignments for them.

A triggering mechanism is concerned with conditions under which changes are to be made to the VP topology and capacity allocations. A design algorithm addresses the question of what changes should be made. Finally, implementation deals with how the changes should be put into place.

2.2.1. Capacity Reallocation

Capacity allocation can be triggered: (a) on demand; (b) when some threshold of spare capacity is reached; (c) when performance monitoring indicates a need for reallocation; (d) when failures require transfer of capacity.

The on-demand approach is used by Ohta and Sato [4] to trigger capacity increase on a set of VPs whenever a new connection cannot be accommodated. In general, if

capacities can be altered quickly enough, on-demand capacity allocation can lead to high network efficiency, as network resources are only reserved as needed. However, due to the overhead associated with a capacity allocation (i.e., processing, bandwidth table updates, possible scheduling changes), it may not be cost-effective to perform reallocations as frequently as connection setups. For the sake of both cost and stability, it may be preferable that the time between successive capacity reallocations be much longer than the time between connection requests.

Additional capacity may be requested when the amount of unused capacity on a VP falls below some threshold value. This method attempts to react to traffic fluctuations before blocking occurs, with the advantage that connection setup need not be delayed. The disadvantage is a decrease in network efficiency, since VPs may be allocated more resources than they require.

A third approach to trigger capacity reallocation is to use information from performance monitoring or forecasting activities. If the average packet delay is monitored, the capacity allocation algorithm can be invoked when performance degrades beyond some operating state. Another possibility is to monitor capacity reservation, capacity usage, and blocked traffic on each VP terminator. Shioda proposes collecting information on call blocking rates [13, 21].

When thresholds or monitoring information are used, capacity reallocation may be done either on a periodic basis (where changes only take place at specific intervals) or in interrupt mode (where network conditions can trigger reallocation at any time).

Changes to the capacity of a VP can be made in quanta (i.e., steps) or calculated continuously. When capacity is added and removed in steps, the size of the step must be determined. A small step size leads to more frequent changes, which has an impact on processing load and stability, while a large step size can result in unused spare capacity. The step size has also an effect on the call blocking probabilities experienced by different VPs. If one wishes to achieve fairness in the sense of equal blocking, one may need to use a large step size for heavily used VPs and a small step size for lightly loaded VPs [4]. Instead

of changing the capacities incrementally, capacity assignments can be computed for each VP based on call blocking probabilities. Call blocking probability is also a good criterion for choosing candidates for capacity reduction [13, 21].

Implementing a capacity reallocation involves changing b andwidth tables used by the CAC algorithm, as well as the parameters of any policing and shaping algorithm that may be applied per VP.

2.2.2. Topology Reconfiguration

The design goal of a topology reconfiguration is to decrease the amount of processing at transit nodes. This is accomplished by providing shorter paths (i.e., direct routes) for source-destination pairs with high traffic demand. A centralized network management facility responds to network congestion by identifying congested pairs of nodes and initiating a topology reconfiguration.

A basic approach is to attempt to reduce the overall VP hop count, which has a direct affect on VC setup. Usage patterns are monitored in order to learn which end-to-end paths become more heavily used so that the number of VP hops can be decreased for these paths. At given intervals, the VP layout is reassessed to determine whether or not VP additions and removals would be beneficial.

The first phase in implementing a topology change is to place the new VPs in the affected routing tables. Then a VP setup policy determines when these VPs should become usable, that is, when capacity can be allocated to the new VPs, and the CAC and route selection algorithms are made aware of them. The second phase of implementation is to terminate or reroute VCs from the old VPs onto the new ones. The third phase is to tear-down the old VPs [20].

2.2.3. Global Reconfiguration

Global reconfiguration is useful when modifications are periodically made based on anticipated traffic conditions (e.g., hourly or daily [13, 21]). It is also useful in facilitating recovery from component failure. Logothetis and Shioda propose an activity where global capacity reallocations are made (without accompanying topology changes) when the observed call blocking probability exceeds some threshold [22].

It is important to note that there is a trade-off between a low setup cost, which favors a low VP hop count, and efficient resource usage, which favors a higher VP hop count. Consider, for example, the use of a VP overlay network consisting primarily of short VPs. This has the effect of producing: (a) a larger number of VPs in the network; (b) VC routes with higher VP hop counts; (c) a larger number of VP terminators located internal to VC routes. The results are higher costs for VC setup and VP maintenance. However, a greater multiplexing gain, therefore lower resource costs, and increased flexibility in establishing VCs are also achieved.

In short, the provision of multiple VPs to enhance conn ectivity and improve fault tolerance, the limitation of the number of VP hops allowed on a route to reduce the processing costs, and the improvement of multiplexing gain to reduce costs are addressed several times in the literature.

The capacity allocation problem generally focuses on performance, that is, minimizing the VC blocking rate or maximizing the carried load. Network adaptability is affected by capacity allocation decisions because traffic conditions can change in unexpected ways. A good allocation of capacity should also allow successive reallocations to be made quickly, and therefore allow more effective adaptation of traffic. For instance, if some capacity on each link is left unassigned (i.e., in a free pool), a capacity increase on one VP can be performed without an accompanying capacity decrease on another VP.

Solutions to the global reconfiguration problem typically involve finding both a VP topology and the capacity allocations for a given traffic scenario, as well as possibly

assigning routes for the VCs. It should be noted that although major changes to the overlay network may take place on a timescale of weeks or months, these changes should be transparent to users in order to meet the requirements of the service providers and their customers [20].

2.2.4. Long-Term Planning

It may be wise to establish a permanent set of VPs that will ensure connectivity regardless of the successive and global modifications that take place. A general VP topology that meets certain criteria, such as connectivity and flexibility, may be better suited to a dynamic setting than an optimized topology based on a specific traffic scenario.

While the goal of global and successive modifications is often optimizing performance for a given traffic scenario, the design of a general or static overlay network may be more concerned with limiting worst-case performance for any traffic scenario, and offering flexibility so that subsequent global or successive modifications (adapting to traffic variations) are easily made.

The characteristics of a "good" general VP topology are considered to be: (a) a low VP hop count between each source-destination pair; (b) a short physical route between each source-destination pair; (c) a low load on the routing tables (i.e., a small number of VPs sharing any given link).

A static layout is created where the worst-case hop count is minimized. Besides providing a static set of VPs, the network could also assign a static minimum capacity to these VPs (or, in fact, to any VP in the network). Each VP is assigned an initial reserved bandwidth, which also serves as its minimum bandwidth [20].

3. FORMAL MODEL AND PROBLEM STATEMENT

As mentioned earlier, the mere existence of VPs improve the network performance in terms of call setup, cell processing time, adaptability and administration. However, if the routing of the VPs and the capacity assignment issues are not handled sufficiently enough to optimize the network performance, these advantages lose their importance against the negative effects of VPs on capacity sharing, call blocking, processing load, and throughput. Therefore, an efficient method has to be developed so that an optimal VP network design can be achieved.

Two possible VP assignment cases can be considered to illustrate the effect of the VPs on the network performance. First, consider a fully connected VP network. A VP network can be formed such that every node has one or more VPs to every other node. In this network, connections are established using only one VP, thus minimizing control costs and the delay encountered during VP switching. But the proliferation of VPs has a negative effect on the network utilization. For most network topologies, the reduced throughput will probably be prohibitive. Second, consider the case where the VP network is the same as the physical ATM network. In this case, the VP network is such that each physical link in the ATM network contains only one VP, which carries all connections in the link. This is equivalent to not using the VP feature at all. In this case, the network utilization is maximized, and the traffic will observe the lowest connection blocking probability. However, the connection establishment activity and processing at the intermediate nodes are at their highest cost. In general, an optimum VP network solution lies between these two extreme cases [12].

The extent to which VP provisioning is able to improve the network efficiency is highly dependent on its ability to provide VCs with low setup and switching costs, while maintaining a low call blocking probability. This, in turn, is dependent on the VP topology and capacity allocations resulting from resource management decisions. In general, the switching cost of a VC increases with the physical hop count and the number of VPs traversed. As for the setup cost, a number of scenarios may arise when a VC setup request is made.

(A) If a VP route is available with sufficient capacity to accept the new VC, setup cost is relatively low, since decisions are made exclusively by the CAC and routing algorithm.

(B) If a VP route exists, but does not have adequate capacity to accommodate the new VC, then the only way to avoid call blocking is to allocate additional capacity to the set of VPs along the route.

(C) Finally, if no VP route is found from source to destination, then the only way to avoid call blocking is by setting up a new VP or a set of VPs.

The last two scenarios incur higher cost because the CAC and routing algorithms cannot establish the connection on their own. It is also not guaranteed that the desired change to the VP topology or capacity allocations can be made. A well-designed VP overlay network tries to maximize the probability of being able to accommodate new VCs without accompanying changes to the VP topology or capacity assignments. Direct VPs or routes with low VP hop counts may be preferred [20].

Based on these observations, this study proposes a general design model for VP routing to exploit the benefits of the VP concept as much as possible without letting its disadvantages be prohibitive for the network performance. The algorithm tries to find the optimal VP layout according to the selected cost function and QoS requirements. It is an asynchronous algorithm in the sense that it works off-line, i.e., it is designed for use at a central point for the whole network and the solution is not real-time. Thus, the algorithm can be computationally complex. As for the strategy the proposed method relies on, it can be classified as a medium-to-long term activity, trying to reply to VC setup requests in the order mentioned above such that the setup costs are kept to a minimum.

3.1. Definitions

This section provides brief explanations about some of the important terms covered by this study. Since detailed descriptions of these terms are beyond the scope of this work, the reader should check the references for more information.

3.1.1. Statistical Multiplexing

Since the VP concept inherently increases call blocking as a result of decreased capacity sharing, it is of particular importance to take statistical multiplexing effects into account. In particular, the bandwidth of a VP can be shared between end-to-end connections via: (a) the establishment/release of end-to-end connections; (b) a bandwidth management scheme which takes into account the variability of traffic on a connection.

The main difference between these two methods is the time scale at which they operate. Indeed, the former method operates at a larger time scale than the latter, which takes into account the micro-variations of traffic at the cell or burst/block levels. Dynamic bandwidth sharing via establishment/release of connections is a classic concept in telephone networks. Statistical multiplexing at the cell or burst level during the connection lifetime is a powerful feature of ATM, inherited from packet switching networks [23].

In statistical multiplexing, the cells of all traffic streams are merged into a queue and transmitted on a first-come-first-serve basis [24], so that the entire capacity is allocated to the first cell out of the queue. The result is a smaller average delay per cell. In other words, in an ATM network, several sources are combined on a single link. When several connections are active on that link at the same time, the required total bandwidth is less than the mathematical sum of the individual connections. The statistical multiplexing gain is determined by the acceptable cell loss rates of the connections [25].

3.1.2. Quality Of Service

ATM applications request particular traffic and QoS parameters when establishing VCs. These requirements consist of bandwidth, latency, and cell-loss values that are itemized by a long string of attributes including peak cell rate, sustained cell rate, maximum burst size, cell transfer delay, cell delay variation, and cell loss ratio.

To help manage the growing complexity of specifying and routing on QoS, the ATM Forum's UNI 3.1 signaling specifies four separate QoS classes that describe general

profiles for QoS parameters. These classes are CBR services, used for circuit emulation; real time variable bit rate services, used for packet audio/video; non-real time variable bit rate services, used for packet data service; and unspecified bit rate (UBR) services. The UNI 4.0 specifies a fifth class, available bit rate (ABR). The UBR and ABR classes represent bursty computer data.

3.1.3. Connection Admission Control

CAC is a design concept which negotiates traffic descriptions between user and network and reserves bandwidth for VC/VP connections to guarantee QoS [26]. CAC is the set of algorithms that determine how many VCs can be carried on a trunk while satisfying QoS requirements. ATM VC routing associates dynamic resource metrics with each trunk that is reduced by each incremental VC supported. The CAC algorithm performs this resource accounting and informs the VC routing algorithm whether an incremental VC can be routed over a particular link in a connection. For ATM networks to be economically viable, CAC algorithms must deliver statistical multiplexing gain; the greater the benefit of statistical multiplexing achieved by CAC, the more VCs a trunk can carry.

In ATM networks, CAC is carried out the following way: When a new call arrives at a local switch, the total capacity of the new call plus existing VCs accommodated by the VP is calculated and compared to the unused bandwidth of that VP. The calculation is performed based on a set of parameters which represent the cell arrival process during the call, thus enabling the evaluation of cell multiplexing characteristics. When the call is accepted, the network verifies the required bandwidth for the call [8].

A common CAC method is equivalent bandwidth CAC. With this method, the signaled traffic parameters and QoS are converted to an equivalent bandwidth for the connection. This value is compared to a trunks unreserved bandwidth to see whether the circuit can be supported. Distributed-routing algorithms can deliver much more dynamic QoS-aware route information since the QoS link accounting is performed locally.

3.1.4. Usage Parameter Control

Usage parameter control (UPC) represents the set of actions taken by the network to monitor user traffic and control (accept/discard) cell traffic to guarantee network QoS. The UPC function is performed at the UNI on the basis of the parameters implicitly or explicitly declared by the user in the end-to-end connection traffic contract. After VP CAC is performed on these declared parameters to check whether there are enough resources available, the UPC function is utilized during the communication phase in order to monitor and control user traffic so as to assure it conforms to the values negotiated at the CAC-phase [26]. In cases where the calls do not conform to the declared parameters, the network exercises a policing function and imposes restrictions on the signal source as necessary [8]. In combination with the CAC, UPC can subsequently prevent any congestion due to traffic contract violations. The UPC function should satisfy these requirements: (a) simplicity of implementation; (b) rapid response to parameter violations; (c) deterministic cell traffic monitoring capability; (d) least probability to accept violating cells or discard conforming cells [25, 26, 27].

3.2. Mathematical Formulation of the Optimization Problem

The network topology is modeled by a directed graph $G = (V,E)$, where V is the set of nodes and E is the set of physical links. G is strongly connected, i.e., there is a directed path from any node to any other. A physical link is defined by its start node, end node and its capacity. A VP is defined by a starting node and an ending node, a directed route between them, and a capacity assigned to this connection. A VC is defined by its start node, end node, and the traffic demand between these nodes in form of a connection metric vector, which are explained later in this chapter. The actual route of a VC, which is a combination of one or more VPs, also belongs to the VC definition.

The physical network topology with its nodes, links and link capacit ies make up the first set of input parameters for the VP routing problem. Another input parameter is the traffic demand matrix in terms of capacities for all node pairs. Given these input

parameters, the proposed design model should provide the set of VPs with their routes, i.e., start, intermediate, end nodes and allocated capacities. It is also the task of the model to determine the combination of VPs to be assigned in order to route the VCs. The notation, the objective function and the constraints are explained later in this chapter.

No restrictions are imposed on the topology of the network. The nodes of the network are functionally identical, i.e., they are not classified as ATM cross-connect or ATM switching nodes. All nodes can be the source or destination of the network traffic. The network links are assumed to be unidirectional. The physical lengths of links are not modeled. These assumptions on the node characteristics are adopted in order to provide an unrestricted solution space for the VP routing problem.

VPs are assumed to be unidirectional logical links which can be established between any two nodes of the network. There cannot be more than one VP with the same endpoints. VPs are assumed to have deterministic bandwidths that are not subject to statistical multiplexing with cells from different VPs. Statistical multiplexing between VCs within the same VP is allowed. All VCs are routed entirely on VPs, i.e., there are no individual VCs carrying traffic without being assigned to a VP.

The processing delay of a cell is represented by the VP hops it encounters. Other factors concerning delay (propagation delay, for example) are not included in the delay representation since they are negligible when compared to the cell processing time at VP exchange nodes because of the updates in the VP routing tables.

3.2.1. Notation

To formulate the optimization problem mathematically the following notation is used throughout this study.

b_k	Mean of the burst period of the k^{th} connection on a VP
C	Set of all VCs
C_{ij}	Set of all VCs passing through l_{ij}

c_{ij}	VC with source-destination pair i and j, $(i,j \in V)$
c'_k	Equivalent capacity of the k^{th} connection in isolation on a VP
C'	Equivalent capacity of n connections multiplexed on a VP
$C'(F)$	Fluid-flow approximation to C'
$C'(S)$	Stationary approximation to C'
d_{ij}	Physical capacity of l_{ij}
E	Set of all network links
h	Maximum number of VP hops
L	Connection metric vector of a VP
l_{ij}	Link connecting adjacent nodes i and j, $(i,j \in V)$, $(l_{ij} \in E)$
m	Mean aggregate bit rate of n connections multiplexed on a VP
m_k	Mean of the peak rate of the k^{th} connection on a VP
n	Number of multiplexed connections on a VP
P	Set of all VPs
P_{ij}	Set of all VPs passing through l_{ij}
p_{ij}	VP with source-destination pair i and j, $(i,j \in V)$
$R_{peak,k}$	Peak rate of the k^{th} connection on a VP
t_{ij}	Traffic demand of a VC c_{ij}
u_{ij}	Link utilization of l_{ij}
V	Set of all network nodes
x	Buffer size of the sources
β	Connection characteristics parameter
σ^2	Variance of the aggregate bit rate of n connections multiplexed on a VP
σ_k^2	Variance of the peak rate of the k^{th} connection on a VP
ε	Maximum cell loss probability (buffer overflow probability)
ρ_k	Source utilization (fraction of active time of the k^{th} connection on a VP)

3.2.2. Objective Function

The objective function to be minimized in this study is the maximum link utilization of the network, which is an important aspect in the design and routing of VPs, since all network traffic is routed over them. The homogenous distribution of the link

utilizations and dynamic VP bandwidth control absorb the effects of traffic imbalances in the network. This effectively improves throughput and the robustness of the network against unexpected traffic conditions [13]. It is obvious that ATM network operation should be robust and flexible enough to overcome the problem of diverse QoS requirements, rather than being optimal for one situation but inflexible for the rest [28]. The minimization of the maximum link utilization gives the network the necessary robustness and flexibility. The objective function can be formulated as:

$$\min\left(\max\left(\frac{\sum_{P_{ij}} C'}{d_{ij}} \right) \right), \text{ for all } l_{ij} \in E \tag{3.1}$$

Within foreseeable future the availability of links that work at a speed much higher than the 10-20 Gb range cannot be expected since the limiting factor for switching must be taken into account. With increasing user access speeds, such as 45 or 155 Mbps, it does not take too many connections to saturate a link that works in the gigabit range. Even in the hypothetical case of practically unlimited bandwidth, it is important to distribute traffic in a way that reduces the maximum link utilization in order to increase network robustness. Clearly, the higher the maximum load on any specific link in the network, the more catastrophic may be the effect of the failure of a link carrying a potentially very large number of connections [6].

3.2.3. Constraints

The physical limitations are natural constraints in the optimization procedure. The sum of capacities allocated to VPs that are using physical link i cannot exceed the physical link capacity C_i. This constraint can be formulated as:

$$\sum_{P_{ij}} C' \leq d_{ij}, \text{ for all } l_{ij} \in E \tag{3.2}$$

The cell loss probability is another requirement to be satisfied in the design of the VP layout and represents the QoS constraints in ATM networks. Principal characteristics of multimedia ATM leased line services provide guaranteed QoS with an end-to-end cell loss rate of at most 10^{-8}. This cell loss rate is guaranteed even while ensuring statistical multiplexing gain for VPs. Thus, when VP connections are accepted, guaranteed QoS is provided and network reactive congestion schemes are not required for VPs. Conservation of the cell loss rate is one of the basic requirements for a good VP accommodation design [26]. In this study, the terms cell loss and buffer overflow are used interchangeably and in order to keep their probability below a given value, the equivalent capacity concept is applied, which will be explained later in this chapter. The cell loss probability is a critical QoS constraint. It is easily converted to capacity requirements and, under certain assumptions, provides a basis for satisfying call blocking constraints with a classical Erlang-B formula [16, 17].

For the delay time requirement, the number of VPs traversed by a VC is used as a constraint in the optimization problem. The rationale behind this limitation is that the VP hop count is related to the number of VP switching nodes on the route of a connection and represents the connection setup times for all VCs. To reduce complexity, VP topologies are composed entirely of direct VPs between node pairs, or routes with $h \leq 2$ in the literature [11, 16, 17, 20, 29]. Since the equivalent bandwidth approach does not take the delay times into account, the number of VP hops is a restricting factor in determining a VCs route.

3.3. Capacity Allocation

The terms capacity and bandwidth are often used interchangeably and can be applied to two levels of resource allocation.

(A) At the individual VP level, capacity allocation refers to the CAC algorithms task of determining how much capacity a given VC will require in order to meet its desired QoS while maintaining the QoS offered to other VCs. Research into this problem has

focused on calculating the effective bandwidth or equivalent capacity of a VC. This level is called VC capacity allocation.

(B) At the network level, capacity allocation refers to the VP management task of determining how network capacity is to be shared between VPs. This level is called VP capacity allocation.

In ATM networks, at the VP level, the capacities of the VCs can be defined deterministically or statistically. Deterministic capacity can be specified by the peak cell rate, which is never allowed to exceed, so that the sum of the assigned VC capacities is never greater than the total VP capacity. With statistical multiplexing, the capacity of a connection is defined by a set of parameters, such as peak/average cell rates and the required QoS. If the parameter values are within the specific path values, the specified cell transmission quality is guaranteed. Individual VCs are allowed to occasionally exceed their peak rates, as long as they conform these parameters. This approach reduces the required total path and link capacity because of the statistical multiplexing effects between VCs in paths. Therefore, if statistical multiplexing is to be implemented at multiplexing nodes, it is necessary to quantify the cell multiplexing characteristics and develop an algorithm that can determine the minimum capacity which still satisfies quality requirements [8].

Several options are available for implementing VP multiplexing and resource reservation. One extreme is to use statistical multiplexing and complete sharing at each link in the network, mixing all traffic types. This method should provide the best multiplexing gain. It also has the advantage that changes to VP capacity assignments do not require modifications to buffer allocation and scheduling at transit nodes. However, QoS guarantees may be difficult to provide, and the CAC algorithm may be complex because heterogeneous traffic sources are multiplexed together at the cell level. The other extreme is to implement deterministic multiplexing and resource reservation at each link in the network with each VP carrying only one class of service. This would simplify CAC and the provision of QoS guarantees, but efficiency may suffer due to peak rate allocation and resource reservation [20].

Because of the statistical multiplexing of connections and shared buffering poin ts in the network, capacity reservation is based on some aggregate statistical measures matching the overall traffic demand rather than on physically dedicated bandwidth or buffer space per connection. In addition to the inherent complexity of such a matching, another major challenge is to provide these control functions in real-time, upon the arrival of a connection request. The corresponding procedures must, therefore, be computationally simple enough so their overall complexity is consistent with real-time requirements.

When connections are statistically multiplexed, their aggregate statistical behavior differs from their individual statistical representation. Therefore, new metrics are needed to represent the effective bandwidth requirement of an individual connection as well as the total effective bandwidth requirement of connections multiplexed on each link. The purpose of the equivalent capacity expression is to provide a unified metric to represent the effective bandwidth of a connection as well as the effective aggregated load on network links at any given time [15].

3.4. Equivalent Bandwidth

The equivalent bandwidth of a set of VCs multiplexed on a VP is defined as the amount of bandwidth required to achieve a desired QoS. In order to characterize the equivalent bandwidth or effective bit rate of VCs in terms of known parameters, an appropriate model such as the statistical characteristics of the VPs at cell level is needed [16, 17]. The equivalent bandwidth is computed from the combination of two different approaches, one based on a fluid-flow model, and the other on an approximation of the stationary bit rate distribution. These two approaches have been selected as they complement each other, capturing different aspects of the behavior of multiplexed connections, while remaining computationally simple. It should be emphasized that the approach has been to develop a simple and yet a reasonably accurate metric to compare load levels on network links, rather than trying to apply exact but intractable models that do not necessarily capture all the impact of complex network interactions [15].

3.4.1. Traffic Model

A single connection has a variable bit rate bounded by its maximum value. In order to characterize the effective bit rate or equivalent capacity of a connection, an appropriate model should be selected to specify its characteristics in terms of known parameters or metrics. A two-state fluid-flow model that captures the basic behavior of the data source associated with the connection is adopted. The rationale for such a model is that a source is either in an "idle state", transmitting at zero bit rate, or in a "burst state", transmitting at its peak rate. Such a source model has the advantage of being both simple and flexible as it can be used to either represent connections ranging from bursty to continuous bit streams or approximate more complex sources.

Based on this two-state fluid-flow model, idle and burst periods are defined to be the times during which the source is idle or active, respectively. The peak rate of a connection $R_{peak,k}$ and distributions of idle and burst periods completely identify the traffic statistics of a connection. Assuming the parameters of a connection are stationary, its peak rate $R_{peak,k}$ and utilization ρ_k, i.e., fraction of time the source is active, completely identify other quantities of interest such as mean m_k and variance σ_k^2 of the bit rate. For exponentially distributed burst and idle periods, the source is furthermore completely characterized by only three parameters, namely $R_{peak,k}$, ρ_k, and b_k, where b_k is the mean of the burst period. The mean burst period b_k gives some indications on how data is being generated by the source. Two sources, with identical mean and peak bit rates but different burst periods, have different impacts on the network. The connection metric vector ($R_{peak,k}$, ρ_k, b_k) represents the most significant aspects of a source behavior.

The equivalent capacity of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired QoS, e.g. buffer overflow probability (cell loss probability), given the offered aggregate bit rate generated by the connections. It is a function of individual connection characteristics and available network resources such as buffers.

The equivalent capacity, therefore, provides a unified metric for link utilizations, which can then be used by network control functions such as routing and congestion control. Because of the "real-time" requirements of these functions, it is critical that the complexity of the equivalent capacity computation be kept as low as possible while still accounting for connections characteristics, existing network traffic, and desired QoS. Exact solutions are either intractable or, when available, incompatible with real-time requirements. As the goal is to provide a simple while still reasonably accurate metric to measure and compare link utilizations, approximations must be developed.

The first approximation accurately estimates the equival ent capacity when the impact of individual connection characteristics is critical. The second approximation is representative of bandwidth requirements when the effect of statistical multiplexing is of significance. As both aspects are typically exclusive, the two approximations complement each other and can be combined to predict relatively accurately the equivalent capacity of connections.

3.4.2. Fluid-Flow Approximation

The first approximation for the equivalent capacity is based on a fluid-flow mode l. In such a model, the bit rate generated by a number of multiplexed connections is represented as a continuous flow of bits with intensity varying according to the state of an underlying continuous-time Markov chain. The aggregate bit rate is offered to a buffer which is emptied at a constant rate. The task is to determine the smallest bit rate C' that, for a given buffer size x, ensures a buffer overflow probability smaller than ε. The value C' is the equivalent capacity of the multiplexed connections. The determination of C' requires that an expression giving the distribution of the buffer contents as a function of the connections characteristics and the service rate be obtained first. This expression must then be inverted to determine the value of the service rate, which ensures an overflow probability smaller than ε for the available buffer size x. This value is the equivalent capacity that should be allocated to the connections.

In the case of a single two-state Markov source, the capacity to be allo cated to the associated connection in isolation has to be determined. Using the notation introduced earlier, a two-state Markov source is characterized by its peak rate $R_{peak,k}$, utilization ρ_k, and mean burst period b_k. Assuming a finite x, the equation satisfied by the equivalent bandwidth for ε (the desired QoS) is then easily found to be of the form:

$$\varepsilon = \beta \exp\left(-\frac{x\left(c'_k - \rho_k R_{peak,k}\right)}{b_k\left(1-\rho_k\right)\left(R_{peak,k} - c'_k\right)c'_k} \right) \tag{3.3}$$

where

$$\beta = \frac{\left(c'_k - \rho_k R_{peak,k}\right) + \varepsilon\rho_k\left(R_{peak,k} - c'_k\right)}{\left(1-\rho_k\right)c'_k} \tag{3.4}$$

Note that an infinite buffer system satisfies the same equation, only with a different value for β.

The equivalent capacity for a single source can then be obtained by solving for c'_k in Equation 3.3. It should be noted that, even for the simple single-source case, no explicit expression is available for the equivalent capacity, and Equation 3.3 must be solved numerically. However, a natural simplification is available, as the term β can be shown to be typically close to 1 (in fact, always smaller). Approximating β by 1 in Equation 3.3 provides an explicit upper bound for c'_k which, furthermore, is close to the exact value. The equivalent capacity associated with a single connection in isolation is then taken to be:

$$c'_k = \frac{\alpha b_k\left(1-\rho_k\right)R_{peak,k} - x + \sqrt{\left(\alpha b_k\left(1-\rho_k\right)R_{peak,k} - x\right)^2 + \left(4\alpha b_k\rho_k\left(1-\rho_k\right)R_{peak,k}\right)}}{2\alpha b_k\left(1-\rho_k\right)} \tag{3.5}$$

where

$$\alpha = \ln\left(\frac{1}{\varepsilon}\right) \tag{3.6}$$

Note that in the case of a continuous bit stream connection $\rho_k = 1$ and $b_k = \infty$, and taking limits in Equation 3.5 yields the expected result $c'_k = R_{peak,k}$.

In the case of multiple superposed sources, where $1/\lambda_k$ and $1/\mu_k$ are the mean idle and burst periods of the k^{th} connection, respectively, and $R_{peak,k}$ is the corresponding peak rate, the value of the equivalent capacity $C'(F)$ given by the flow approximation for n multiplexed connections is defined by:

$$C'(F) = \sum_{k=1}^{n} c'_k \tag{3.7}$$

where c'_k values are determined from Equation 3.5 and the following relations are used:

$$\mu_k = \frac{1}{b_k},\ \lambda_k = \frac{\rho_k}{b_k(1-\rho_k)},\ \text{and}\ \lambda_k + \mu_k = \frac{1}{b_k(1-\rho_k)} \tag{3.8}$$

The major advantages of Equation 3.7 are its computational simplicity and its explicit dependency on source parameters. However, although the linearity of Equation 3.7 certainly simplifies the accounting of how bandwidth is allocated to connections, it also clearly identifies the limitations of the expression. The linearity explicitly indicates that the simplifying assumption $\beta \cong 1$ amounts to ignoring the effects of statistical multiplexing. In particular, unless the equivalent capacities of individual connections are themselves close to their mean bit rates, their sum is typically an overestimate of their equivalent capacity. Another approximation is, therefore, needed to accurately determine the required bandwidth allocation for cases in which statistical multiplexing is significant.

3.4.3. Stationary Approximation

The base for an approximation, when the effect of statistical multiplexing is the dominant factor, can be obtained by studying the impact of the assumption $\beta \cong 1$. Turning again to the case of n identical two-state Markov sources, β is significantly different from 1 when a number of connections with equivalent capacity much larger than their mean bit rates are multiplexed. This is essentially the case for connections with long burst periods and relatively low utilization.

More specifically, the value *C'(S)* of the equivalent capacity can be selected to ensure that the aggregate stationary bit rate exceeds *C'(S)* only with a probability smaller than ε, the desired buffer overflow probability. This clearly ensures a buffer overflow probability below ε, but is often a substantial overestimate of the actual bandwidth required as it ignores the "smoothing" effect of the buffer, i.e., the buffer allows the input rate to exceed the output rate for "short" periods. In the case of connections with long burst periods, it can, however, be argued that because the aggregate bit rate remains at the same value for relatively long periods, overload situations (when present) are likely to last sufficiently long to result in buffer overflow. It is then reasonable to allocate enough bandwidth to make the probability of an overload condition equal to the desired buffer overflow probability.

The assumption of a Gaussian distribution allows standard approximations to be used to estimate the tail of the bit rate distribution. In particular, the value *C'(S)* should be determined such that the cumulative tail probability beyond *C'(S)* does not exceed ε. The value of *C'(S)* can then be obtained from approximations for the inverse of the Gaussian distribution. A good approximation is given by:

$$C'(S) \cong m + \alpha'\sigma \tag{3.9}$$

with

$$\alpha' = \sqrt{-2\ln(\varepsilon) - \ln(2\pi)}\,,\ m = \sum_{k=1}^{n} m_k\,,\ \text{and}\ \sigma^2 = \sum_{k=1}^{n} \sigma_k^2 \qquad (3.10)$$

where m is the mean aggregate bit rate, and σ is the standard deviation of the aggregate bit rate. Note that Equation 3.9 is also computationally simple and only depends on the QoS and the means and variances of the bit rates generated by individual connections. These are directly available from the source characteristics defined earlier.

3.4.4. Derivation of Equivalent Capacity

It remains to combine the flow and stationary approximations given in Equatio ns 3.5, 3.7, and 3.9 into a single expression giving the equivalent capacity of a set of connections. As both approximations overestimate the actual value of the equivalent capacity and are inaccurate for different ranges of connections characteristics, the equivalent capacity *C'* is taken to be the minimum of *C'(F)* and *C'(S)*. That is:

$$C' = \min\{C'(F), C'(S)\} \qquad (3.11)$$

It should be mentioned that although Equation 3.11 is the minimum of the values obtained from both the flow and stationary approximations, it still overestimates the actual bandwidth requirements. This overestimation is, however, typically reasonable. Furthermore, a somewhat conservative allocation is preferable to underestimating the bandwidth requirements of connections, which could result in network congestion or even failure.

3.4.5. Connection Metrics

Based on the expression given in Equation 3.11, it is possible to define connection metrics that characterize the current capacity allocation on network links and VPs. This metric must provide a simple and compact form to store the bandwidth allocation

information, while allowing for real-time updates and computations of allocation levels. A three-dimensional vector representation meeting these requirements is:

$$L = \left(m \quad \sigma^2 \quad C'(F)\right) \tag{3.12}$$

where L is the connection metric vector on a VP, and as previously defined, m and σ^2 are, respectively, the mean and variance of the aggregate bit rate corresponding to all the connections routed on this VP, and $C'(F)$ is the sum of their individual equivalent capacities.

A major aspect of Equation 3.12 is the "incremental" property of the connection metric vector. In particular, updates can be performed based solely on the combined information contained in the connection metric vector and the characteristics of the connections being established or released. There is no need for specific information on the individual characteristics of the connections currently routed over the VP. In particular, a connection metric update is executed as follows.

(A) Upon receipt of a connect or disconnect request, the source node associated with the connection computes, from the connection metric vector ($R_{peak,k+1}$, ρ_{k+1}, b_{k+1}), a request vector of the form (m_{k+1} σ_{k+1}^2 c'_{k+1}), where positive values indicate a connect request and negative values a disconnect request.

(B) The request vector is sent to all nodes with links over which the connection is or should be routed, and the corresponding connection metric vectors are updated by performing a simple vector addition/subtraction.

In summary, the problem can be formulated as shown in Figure 3.1.

GIVEN	:	- $G = (V,E)$, the physical network topology with end nodes, links and link capacities; - C, the set of all VCs, i.e., connection requests defined by R_{peak}, the traffic demand;
MINIMIZE	:	- maximum value of u_{ij}, the link utilization on any link l_{ij};
SUBJECT TO	:	- cell loss probability $\leq \varepsilon$; - # of VPs traversed by a VC $\leq h$;
DESIGN VARIABLES	:	- P, the set of all VPs defined by their routes and allocated capacities; - route of each VC in terms of VPs.

FIGURE 3.1. Problem Formulation.

This is a difficult optimization problem. Therefore, a heuristic design algorithm is proposed in the next section.

4. HEURISTIC DESIGN ALGORITHM

In order to solve the hard optimization problem covered in this study, it is necessary to compromise the requirements of systematicity and to construct a control structure that can find a good solution in less than exponential time. Therefore, the design algorithm is based on heuristics, which is a technique that improves the efficiency of a search process, possibly by sacrificing claims of completeness [30]. The method developed in this study defines a search algorithm to check for the optimum solution in the domain of valid VP assignments. The algorithm consists of two basic phases, which are initialization and optimization. In the former, a starting point is found which is a valid solution, i.e., a valid VP network that satisfies the constraints. In the latter, incremental changes are made in the VP network that achieve a lower value for the objective function and satisfy the constraints, until no more improvement can be found.

The pseudo-code for the main part of the algorithm is given in Figure 4.1.

```
Program: VP Design

Phase 1: Initialization
read traffic requirements and convert them to equivalent bandwidths;
create an initial set of VPs for node pairs having a traffic demand and a direct link between them;
try to route all connection requests over these initial VPs;

Phase 2: Optimization
repeat
        sort VCs not assigned yet in descending order according to their capacity requirements;
        for every VC in the list try to find the most idle route by checking the alternatives;
        sort the physical links in descending order according to their utilizations;
        for every physical link in the list
                make a list of VPs on that link;
                repeat for every VP in the list
                        try to find an alternative route with a lower maximum link utilization value;
                until a VP is rerouted successfully or end of VP list;
                if no VPs are rerouted then make a list of VCs on the same link;
                repeat for every VC in the list
                        try to find an alternative route with a lower maximum link utilization value;
                until a VC is rerouted successfully or end of VC list;
until no improvement can be achieved on any link;
```

FIGURE 4.1. Pseudo-Code for the Heuristic Design Algorithm (HDA).

As previously mentioned, there are two extreme cases that give boundaries on the range of possible solutions.

(A) Each physical link is a VP. This case corresponds to maximum sharing. It yields the lowest call blocking probabilities for a fixed connection routing. However, it results in the maximum number of VPs along the path of a connection.

(B) Each connection has its own VP from source to destination. This case corresponds to the minimum processing delay. Conversely, the link sharing is minimized. Each connection has its own capacity along the physical path.

Clearly, the design algorithm should be able to find a high quality solution between these extreme cases, i.e., a VP network design that balances link sharing, blocking probabilities, and processing cost in the best possible way.

4.1. Initialization Phase

In this phase, an initial subset of VPs, P', is established first using the trivial idea of creating a VP p_{ij} on every physical link l_{ij} of the network under the following conditions:

$$t_{ij} > 0, \text{ for all } l_{ij} \in E \tag{4.1}$$

$$d_{ij} \geq t_{ij}, \text{ for all } l_{ij} \in E \tag{4.2}$$

The inequality given in Equation 4.1 implies that there is a traffic demand between the nodes i and j. The inequality given in Equation 4.2 implies that there is sufficient physical link capacity to accommodate the traffic between i and j. Every p_{ij} out of these VPs is assigned automatically to c_{ij} carrying the traffic t_{ij}, which requires a bandwidth c'_k determined by means of the equivalent bandwidth method for a single connection. The maximum number of hops, h, is not a limiting factor yet since all these VCs use only one VP from the source to the destination. The selection order of the VPs is irrelevant since there is no multiplexing of VCs on VPs or physical links yet. The reason for this initial VP

layout where every physical link is a VP is, as mentioned above, making use of direct physical routes.

After this first step, the remaining VCs, i.e., the ones which represent connection requests between nodes without a direct physical link from the source to the destination, are handled. For this purpose, the best combination of existing VPs, i.e., the most idle one in terms of link utilization, is sought after. If no such combination exists, the algorithm looks for the best solution taking its right to creating new VPs into account. The details of this search heuristics are explained later in Chapter 4.3.2.

At the end of the initialization phase, normally a feasible solution, i.e., a VP routing scheme in which all VCs are assigned to some combination of VPs without violating any of the constraints, should appear. If, however, some of the VCs are left unassigned, the algorithm proceeds with the optimization phase because these VCs might still get a chance to be assigned as a result of reallocations.

4.2. Optimization Phase

Once an initial VP layout is designed which serves all the VCs and satisfies the QoS requirements, it is time to start with the optimization activities. In this phase, the main concern is to reduce the link utilization on the most heavily loaded link of the network by applying VP and VC movement activities, which are explained below. The algorithm also tries to reduce the congestion of other links, even if no improvement can be made on the worst loaded link at some point, because these reallocations may lead to free bandwidth that can be used for reallocating VPs or VCs on the worst loaded link in later turns.

The optimization phase starts by sorting all physical links, l_{ij}, in decreasing order according to their congestion ratios, u_{ij}. First, for every physical link l_{ij}, the possibility of moving one of the previously assigned VPs, i.e., a VP from P_{ij}, the set of the VPs using l_{ij}, to other links is checked. Then, the same procedure is applied to the VCs from C_{ij}, the set of the VCs using l_{ij}. The VP movement activity is prior to the VC movement activity

because the statistical multiplexing gain is not lost and no recalculation of the equivalent bandwidth is needed when a whole VP is rerouted, whereas separation of one or more VCs from a VP requires equivalent bandwidth recalculations and may result in loss of multiplexing gain.

The optimization activities are restarted whenever there is some improvement in any of the links utilization u_{ij}. If there are remaining VCs from the previous phase, the algorithm tries to route them first. Then the links are sorted again, and the whole process is repeated. The optimization phase is terminated when no improvement is made for any link in one whole iteration.

4.2.1. Virtual Path Movement

By moving a previously assigned VP from P_{ij}, i.e., changing its physical route, the link utilization u_{ij} of l_{ij} is tried to be reduced. For this purpose, alternate routes connecting the same VP endpoints are checked, and the one with the least maximum utilization is selected. The choice of the best alternate route is based on three key procedures which are explained in detail in Chapter 4.3. The VP with the largest capacity is selected first to be rerouted, because when the VP with the larger bandwidth is first moved to other links, the remaining capacities on these links become small, however, even in that case, there still remains a possibility that the VP with smaller bandwidth can be fit into those links. If the suitable alternate path cannot be found for that VP, the next VP with the largest capacity of the same link is checked. This procedure is repeated this way until all VPs from P_{ij} are checked.

4.2.2. Virtual Circuit Movement

If the VP movement activity on the link l_{ij} yields no result, then the VC movement activity begins on that link. The procedure has the same motivation as the VP movement activity, but this time concerning VCs instead of VPs. In this procedure, one of the VCs using link l_{ij}, i.e., a VC from C_{ij}, is separated to be rerouted over an alternate VP combination. The alternate VPs should have a less heavily loaded physical route and

enough free bandwidth to accommodate the newly coming VC. The statistical multiplexing gain implies that the total required bandwidth is increased when the single VP is separated into several VCs. Also, the recalculation of the equivalent bandwidths on the old and new routes should be considered. This procedure is repeated this way until all VCs from C_{ij} are checked.

4.3. Key Procedures

The algorithm developed in this study uses heuristic methods to achieve a satisfactory VP design. Some of them, which can be seen as the key procedures of the VP routing algorithm, are mentioned above while summarizing the logic behind the phases and activities. These key procedures are briefly explained in the following sections.

4.3.1. Equivalent Bandwidth Calculations

The equivalent bandwidth c'_k of an individual connection c_{ij} has only to be calculated once at the beginning of the algorithm according to Equation 3.5. Its mean m_k and variance σ_k^2 can be calculated using the connection metric vector ($R_{peak,k}$, ρ_k, b_k). They also have to be calculated just once. The equivalent bandwidths as well as the mean and variances of all the connections from C_{ij} are computed by a preprocessor in the algorithm before the initialization phase. This is a strong property of the equivalent bandwidth concept since it does not require recalculations of these individual parameters concerning the connections QoS constraints. In other words, once these values are calculated, the main concern of the algorithm is to map various VPs onto the backbone network and assign them VCs so that the maximum link utilization is minimized.

The equivalent bandwidth of a VP has to be recalculated whenever a change in the VC assignment of the VP occurs. These recalculations , however, are simple additions or subtractions done with the individual connection parameters c'_k, m_k and σ_k^2 according to Equations 3.7, 3.9, and 3.10. The computational simplicity of performing these calculations gives the algorithm the chance to make them repetitively, try various "what if" cases and

select an optimum solution from the solution space without leaving any possibility unchecked.

The amount of the occupied capacity of a physical link l_{ij} is merely the sum of all capacities of the VPs from P_{ij}. It also has to be recalculated when a change in one of the VPs from P_{ij} occurs.

4.3.2. Search for the Best Route

When the algorithm has to find the best VP route for a VC c_{ij} in terms of the congestion experienced in the underlying physical network, there are several possibilities. First of all, c_{ij} might have been assigned to some combination of VPs already. In this case, the task is to find an alternative to the existing route which improves the maximum congestion along the route of c_{ij}. If c_{ij} is not previously assigned to a route of VPs, the procedure only has to find a valid solution set and choose the best candidate among them.

In any case, the procedure examines P, the set of VPs, first, in order to find an existing VP which can carry the traffic of c_{ij}. If there is one, it is a candidate offering a one VP hop. Next, the best candidate for a two hop solution with existing VPs is sought after. Then, other possible combinations are checked. Trying to combine one existing VP, which starts at the source node and ends at a pivot node, and a new VP, which starts at this pivot node and ends at the destination node, and vice versa, may produce the third and fourth candidates. Finally, a completely new VP, connecting the source with the destination directly with one hop, is checked. The creation of a new VP involves decisions about its physical route, which are given by the idlest path procedure explained in Chapter 4.3.3.

Once these five candidates are found, their maximum link utilization values are compared and the candidate offering the lowest value is selected as the best route, which is assigned as the new route for c_{ij}. In the case where two or more of the candidates offer the same amount of improvement, use of existing VPs is always encouraged since it increases capacity sharing and therefore decreases call blocking probabilities.

4.3.3. Idlest Path Technique

The purpose of this procedure is to find the most idle physical route between a given source and destination. It is based on a shortest path algorithm [31] and is modified in such a way that it returns the list of nodes on the most idle path between two nodes and its maximum congestion. The main idea of the procedure is to assign each link a "length" that reflects its current congestion. The new VP is routed along the shortest path with respect to this length between given source and destination nodes. In other words, this procedure is based on a modified shortest path algorithm with three basic steps: (a) initializing the "lengths" of all links connecting the source with other nodes; (b) finding incrementally the "nearest" adjacent node to the current "idlest path tree"; (c) updating the idlest path tree and the "lengths" of remaining vertices.

In the first step, the "lengths" of all nodes other than the source are calculated the following way: if there is a physical link between the source and a node, then the "length" of this connection is the utilization of the link; if not, it is 1, i.e., the maximum utilization allowed. This way, the utilization is used as the cost of reaching from the source to a node. By assigning 1 to connections which do not have actual physical links, the procedure makes sure that some nodes are not reachable from the source.

The second and third steps are repeated after each other until the idlest path tree is completed. In each iteration of these steps, the procedure selects the "closest" node to the idlest path tree, adds it to the tree, and updates the "lengths" of the connections to nodes not yet reached the following way: if the maximum utilization of the current tree is greater than the utilization of a new link, then the "cost" of using that link remains unchanged; if not, the "length" of the new link is replaced by the maximum utilization of the current tree.

During these steps, the procedure also makes a list of the nodes that make up the most idle path for the given source-destination pair. At the end of the procedure, this route information is returned to the calling procedure if it is feasible, i.e., the maximum link utilization of the route is less than 100 per cent. The pseudo-code for the procedure is given in Figure 4.2.

```
Procedure: Idlest Path (Source,Destination;Nodelist)

Phase 1: Initialization
mark Source reached;
for all other nodes
        mark them not reached;
        assign Source as predecessor to them;
        compute utilizations from Source to them;

Phase 2: Search
repeat
        find the node with minimum utilization which is not reached yet;
        for the node found
                mark it current node;
                mark its utilization minimum utilization;
                mark it reached;
        for all nodes not reached yet
                compute new utilization from current node to this node;
                if utilization of this node > max(minimum utilization , new utilization) then
                        utilization = max(minimum utilization , new utilization);
                        assign current node as predecessor to this node;
until all nodes are reached;
create the Nodelist of the physical route by going over predecessors from Destination to Source;
```

FIGURE 4.2. Pseudo-Code for the Idlest Path Procedure.

The intuition behind this model is based on viewing the routing problem as an instance of multicommodity flow problem, which involves simultaneously shipping of several different commodities from their respective resources to their destinations in a single network so that the total amount of flow going through each edge is no more than its capacity. Associated with each commodity is a demand, which is the amount of that commodity that we wish to ship. In the routing concept, commodities correspond to VCs, demand correspond to the requested bandwidth, and capacity corresponds to the link bandwidth. The multicommodity flow solution is optimized if all flows are routed along shortest paths. From the theoretical point of view, this approach is attractive because it unifies the routing and admission control decisions. In other words, there is no need for explicitly controlling physical link capacity limitations since it is embedded in the procedure in a natural way by rejecting solutions with a link utilization greater than 100 per cent as infeasible.

5. COMPUTATIONAL EXPERIMENTS

The heuristic design algorithm proposed in this study uses the equivalent bandwidth concept to guarantee a desired QoS, limits the maximum allowed number of VP hops to meet processing delay constraints and tries to optimize the network performance by minimizing the maximum link utilization under these conditions. There are no computational results or numerical examples provided in the literature achieved by using exactly the same objective function and constraints to test the quality of the proposed algorithm directly. A lower bound for the objective function is also hard to find since the link utilization is not an absolute value but the ratio of the used capacity over the total capacity of the link. Moreover, a network large enough to yield a non-trivial VP system is too large for an exhaustive search unless the number of VCs to be routed is limited, which is not a realistic approach since a traffic matrix is normally not sparse. In a network of four nodes and eight links, where there is only two possible routes for a VC, the solution set consists of 2^{12} different solutions. Therefore, test algorithms have to be developed explicitly to see the quality of the results.

The first competitor, which is called "Idlest Path Routing", is a variation of the proposed heuristic design algorithm itself. As previously mentioned, there are two phases in the original algorithm, which are initialization and optimization. Between these two phases, VCs which are not routed yet are sorted decreasingly according to their bandwidth requirements. The effects of the initialization phase and the sorting of waiting VCs in the original algorithm can be observed by omitting these features. Moreover, the random choice of VCs to be routed may yield to load balancing and, in some cases, Idlest Path Routing can be a good competitor to the proposed heuristic design algorithm. Chlamtac chooses the physical routes of VPs at random from the set of all possible shortest paths [6, 10]. This tends to result in a more balanced use of the network resources. So, a similar result can be achieved through the random choice of VCs waiting to be routed.

The heuristic design algorithm is also compared to statistical quality measures by using two additional competitors which involve different levels of randomness. The first one routes VCs on a randomly selected combination of physical links, disregarding all

constraints concerning QoS like cell loss probability, capacity and delay. The idea is to find some good VC distributions on the physical network layout by trying many different solutions and selecting the best one. Since the competitor ignores all capacity restrictions, it is clear that some of the solutions are infeasible, i.e., they violate these restrictions. Therefore, a second competitor is developed which also routes connections on random paths, but takes the physical capacity limitations into account. In other words, the second competitor tries to route the VCs on a randomly selected path which does not utilize the link beyond hundred per cent and rejects any VC that exceeds this feasibility threshold. Both competitors choose the VCs to be routed in random order.

The heuristic design algorithm is tested regarding four important criteria in the evaluation of a network design methodology. These are network size, network density, traffic type and traffic load. Varying the size and the density of a network gives an idea about the behavior of the algorithm in different physical network topologies. The network size and density are represented by the number of nodes and links in the network respectively. Changing the type and the load of the network traffic shows the quality of the heuristic design algorithm under different traffic conditions. To simulate different traffic types, or patterns, and traffic loads on the network, several distributions and peak rate ranges of connection requests are used in the traffic demand matrices.

5.1. Demonstrative Example

The purpose of this chapter is to show a typical run of the heuristic design algorithm with the input, output and design variables. The model of the demonstration represents a sparse network with 8 nodes and 31 directed links shown in Figure 5.1. There are two types of physical links according to their capacities. These are STS-3 and STS-12 links defined as 155.520 Mbps and 622.080 Mbps, respectively [25].

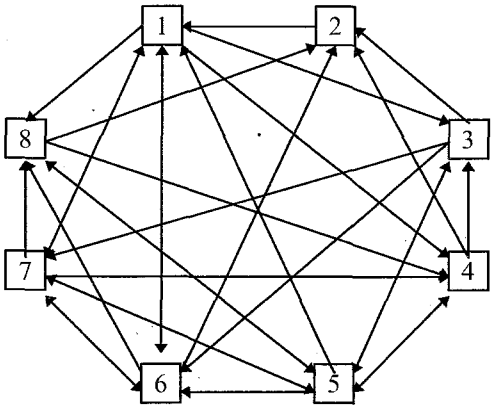

FIGURE 5.1. Physical Network Topology.

The offered traffic load in the network is shown in Table 5.1, which is obtained by scaling the results of a metropolitan area network simulation used by Ryu [16, 17] and Shioda [13] such that the capacity limitations do not become a bottleneck for the demonstration. The table shows the R_{peak} values of the traffic demands.

TABLE 5.1. Offered Traffic Load Matrix (Mbps).

s \ d	1	2	3	4	5	6	7	8
1		85.100	75.820	16.620	13.240	20.780	18.660	26.540
2	79.080		112.500	22.740	18.620	20.640	21.820	19.720
3	75.140	121.940		100.880	75.780	108.160	100.120	63.140
4	18.160	22.640	100.640		51.060	52.860	24.860	12.840
5	15.640	20.540	76.500	55.320		49.420	20.680	11.360
6	20.900	30.420	111.940	61.860	48.820		60.480	20.908
7	19.200	22.320	110.500	22.940	17.320	51.900		41.100
8	26.000	20.400	74.400	13.480	11.700	20.560	51.220	

The heuristic design algorithm is run on the given network topology with the offered traffic load and additional network parameters such as $x=5$ *Mbps*, $\varepsilon = 10^{-5}$, $h=2$, $\rho_i=0.5$ and $b_i=100$ *ms*. It starts by computing the equivalent bandwidths of the individual connection requests form the traffic loads. Then it creates initial VPs on the physical links and routes connecting requests between adjacent nodes on these VPs. After the initialization phase, the algorithm makes 17 iterations in the optimization phase and finishes with a maximum link utilization of 0.493 and an average of 0.342, having created a total of 43 VPs, which are listed in Table 5.2, and routed all 56 VCs over these VPs.

TABLE 5.2. VPs of the Network (bandwidths are in Mbps).

VP #	VP Route (Nodes)	VP Bandwidth
1	1;3	188.310
2	1;4	124.450
3	1;6	114.820
4	1;7	40.470
5	1;8	46.250
6	2;1	276.470
7	3;6;2	121.940
8	3;5	75.780
9	3;6	171.290
10	3;7	100.120
11	4;2	130.050
12	4;3	100.640
13	4;5	88.750
14	5;6;1	15.640
15	5;3	76.500

VP #	VP Route (Nodes)	VP Bandwidth
16	5;4	117.170
17	5;6	49.420
18	5;7	96.750
19	5;8	44.730
20	6;1	39.050
21	6;2	30.420
22	6;7	60.480
23	6;8	244.860
24	7;1	45.190
25	7;4	45.250
26	7;8;5	17.320
27	7;6	51.900
28	7;8	41.100
29	8;2	40.930

VP #	VP Route (Nodes)	VP Bandwidth
30	8;4	13.480
31	8;5	130.340
32	8;4;3	186.330
33	7;5;1;3	110.500
34	3;7;5;1	75.140
35	6;8;5	61.860
36	4;5;1	52.860
37	2;1;8	18.620
38	4;5;7;1;6	18.160
39	1;4;5	13.240
40	8;4;5;7;6	0.000
41	3;7;5;4	100.880
42	8;5;1	20.560
43	8;4;5;7	26.000

The list of VPs shows that the routes of 28 VPs from 31 created in the initialization phase remain unchanged. Twelve additional VPs are created during the optimization phase. The average number of links along the route of a VP is 1.535. In other words, most of the VPs use direct physical routes, which decreases the total resource usage. The reason for this is the creation of VPs on every physical link that has a traffic demand between its end nodes in the initialization phase. In the rest of the optimization process, the algorithm routes, and reroutes, the connections mainly on these VPs and does not need to create many new VPs which could lead to capacity segmentation and waste of unused bandwidth.

The connection requests between each node pair, i.e., the VCs, are accommodated by either one VP or a combination of two VPs regarding the maximum link utilization encountered along the physical routes of each candidate paths. The assignments of VPs to VCs are shown in Table 5.3. Thirty six VCs are routed over single VPs and 20 VCs use routes made of two VPs. The average VP hop count for the VCs is 1.357, and 86 per cent of the assignments are to the VPs created in the initialization phase. This shows that the algorithm tends to use VPs with short physical routes in order to decrease the maximum link utilization, which is logical since short VP routes mean low resource usage as mentioned above.

TABLE 5.3. Assignment of VPs to VCs.

s \ d	1	2	3	4	5	6	7	8
1		2;11	1	2	39	3	4	5
2	6		6;1	6;2	37;31	6;3	6;4	6;5
3	34	7		41	8	9	10	9;23
4	38;20	11	12		13	36;3	13;18	13;19
5	14	19;29	15	16		17	18	19
6	20	21	23;32	35;16	23;31		22	23
7	24	25;11	33	25	26	27		28
8	43;24	29	32	30	31	42;3	31;18	

After routing the VPs on the physical links, every link becomes loaded in the ratio of its used bandwidth allocated to one or more VPs over its total capacity. The resulting link utilizations are shown in Table 5.4.

TABLE 5.4. Utilization of the Physical Links after the VP Routing.

s \ d	1	2	3	4	5	6	7	8
1			0.480	0.221		0.214	0.065	0.417
2	0.474							
3		0.000			0.487	0.471	0.444	
4		0.209	0.461		0.320			
5	0.416		0.492	0.351		0.418	0.226	0.288
6	0.352	0.245					0.389	0.493
7	0.407			0.291	0.461	0.334		0.376
8		0.066		0.363	0.370			

The algorithm is demonstrated with its input, output and design variables for small size networks. In what follows, comparisons are made with the competitors and the same input data to show the quality of the results. This section also gives a preview of the way the heuristic design algorithm, the idlest path routing algorithm and the random solution generators used for the tests are compared in the next chapters.

The competitors are run 10000 times. Every iteration of each competitor gives a result for the maximum link utilization. Table 5.5 displays the results of the random solutions and the idlest path routing solution in comparison with the result of the heuristic design algorithm.

TABLE 5.5. Comparison of the Solutions for the Demonstrative Example.

	Heuristic Design Algorithm	Idlest Path Routing (10000 Runs)	10000 Feasible Random Solutions	10000 Random Solutions
Best Result	0.493	0.553	0.644	0.968
Average Result	0.493	0.741	0.918	1.991
Worst Result	0.493	1.000	1.000	4.192

In Table 5.5, the "Best Result" row shows that the heuristic design algorithm finds a lower value than its competitors for the maximum link utilization. The best result of the idlest path routing algorithm is 12 per cent worse than the one found by the heuristic design algorithm. The difference of the performances is even bigger when the heuristic design algorithm is compared to the random solution generators. The performance of the heuristic design algorithm is better than 10000 feasible random solutions and 10000 random solutions by 31 per cent and 96 per cent, respectively. Another advantage of the heuristic design algorithm in this case is that it can be used synchronously because of its speed. It takes the heuristic design algorithm a few iterations to find the result, while its competitors have to make 10000 consecutive trials to find a good solution.

The performances of the solutions can also be compared by observing their mean values displayed in the "Average Result" row above and the distributions of the 10000 solutions they offer. Figures 5.2, 5.3, and 5.4 display the distributions of the solutions generated by the competitors. The average of the results increases as intelligence is replaced by randomness. The figures show together how the distribution is shifted from smaller utilization values for the idlest path routing algorithm towards larger ones for the 10000 random solutions. When the average values and distributions of the idlest path routing algorithm and 10000 random solutions are compared with the result found by the heuristic design algorithm, which is 0.493, it can be seen that the performance of the latter is 50 per cent and 300 per cent better on the average, respectively.

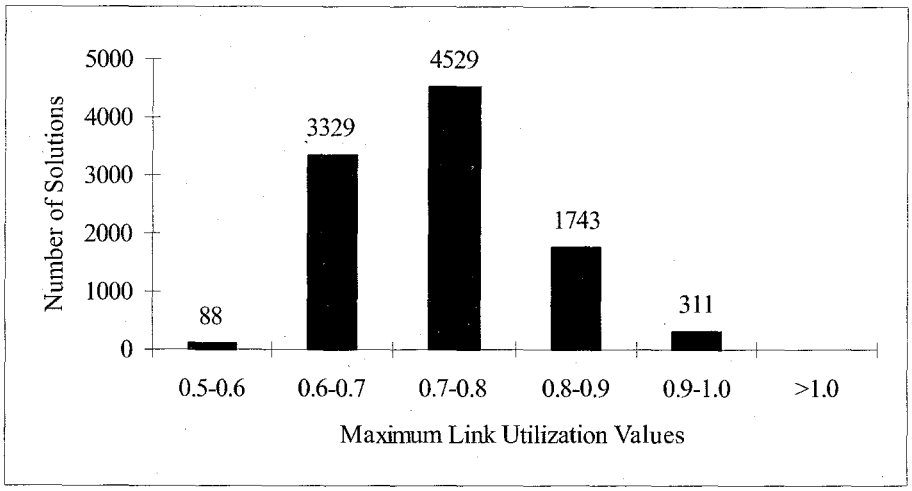

FIGURE 5.2. Distribution of 10000 Idlest Path Routing Solutions.

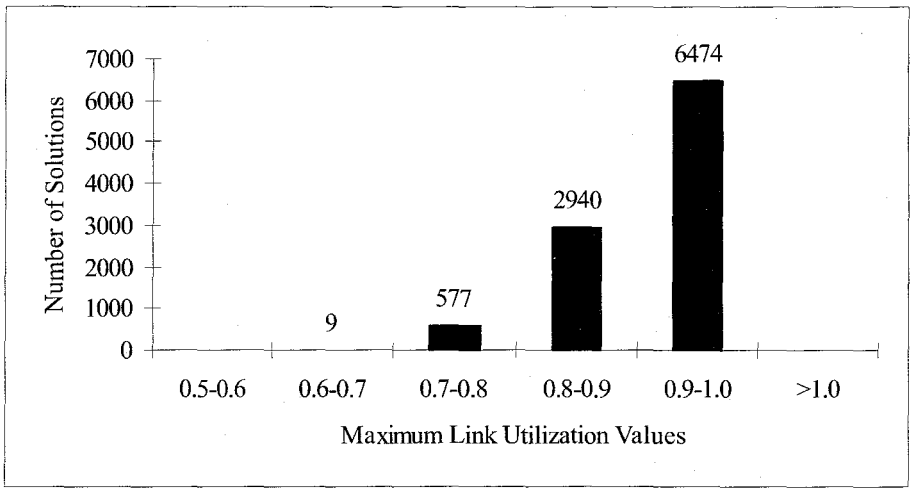

FIGURE 5.3. Distribution of 10000 Feasible Random Solutions.

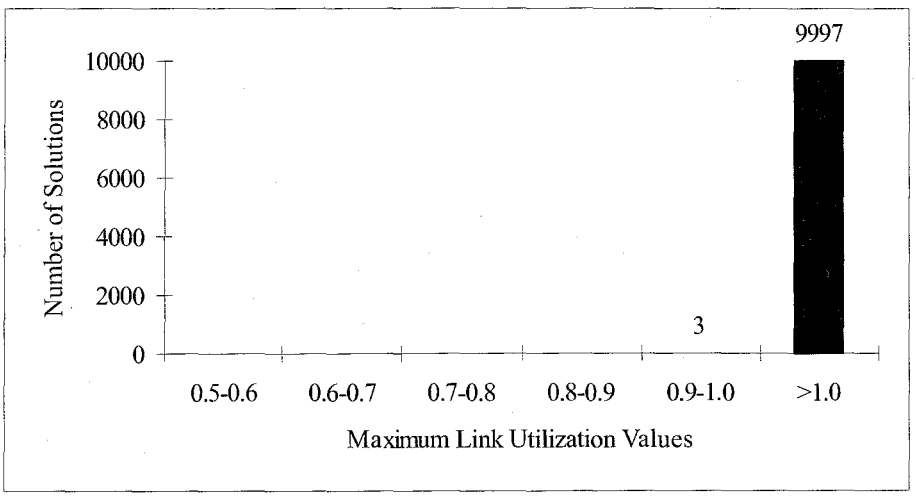

FIGURE 5.4. Distribution of 10000 Random Solutions.

5.2. Tests and Results

The heuristic design algorithm is compared to three different competitors under various network and traffic conditions. Two of them create random solutions and the third is an alternative algorithm based on the heuristic design algorithm itself but handles the VCs in random order. The tests are performed on networks of different sizes and densities, and with different traffic types and loads.

In the experiments, four different size networks with 8, 16, 32 and 64 nodes are used. The number of links used in connecting these nodes to represent sparse and fully connected networks are listed in Table 5.6. A fully connected network consists of direct physical links between any two nodes. On the other hand, a network is sparse if the number of links is less than half of a fully connected network but there is still a physical path between any two nodes.

TABLE 5.6. Numbers of Links defined for Different Numbers of Nodes.

Network Size (# of Nodes)	Sparse Networks	Fully Connected Networks
8	27	56
16	116	240
32	486	992
64	1990	4032

The ranges for link capacities of sparse and fully connected networks are different. In sparse networks, the physical bandwidth of a link varies between 155.520 Mbps and 207.360 Mbps. In fully connected networks, where there are more links and, as a result of this, more total free bandwidth to use, these limits are defined between 103.680 Mbps and 207.360 Mbps, so that the effects of different traffic conditions can be seen more clearly. These values are multiples of STS-1, a standardized high-speed optical link capacity of 51.840 Mbps [24].

Three different types of traffic are used in the network. The first type is uniform traffic, where the demand between nodes is normally distributed. The second type is called centralized traffic, where the demand to and from certain nodes (centers, or servers) is

defined to be higher than the demand between others. Finally, the third type builds communities of interest, where there are user groups in the network. The traffic between members within the same group is defined to be higher compared to the traffic between members from different groups. The characteristics of these traffic types are defined in Tables 5.7 and 5.8.

TABLE 5.7. Definition of the Centralized Traffic.

# of Nodes	# of Centers
8	1
16	2
32	4
64	8

TABLE 5.8. Definition of Communities of Interest.

# of Nodes	# of Groups	# of Members in Each Group
8	2	4
16	2	8
32	2	16
64	2	32

Since the main idea behind centralized and community type traffic simulations is the non-uniform nature of the traffic load distribution, these traffic types are represented by variable traffic demand ranges as summarized in Table 5.9. The values are given for heavy traffic conditions and are halved for a light load of traffic.

TABLE 5.9. Capacity Limitations (R_{peak}) for Connection Requests (Mbps).

Traffic Type		Sparse Networks	Fully Connected Networks
Uniform		0-43.200	0-69.120
Centralized	for center(s)	43.200-86.400	69.120-138.240
	for other nodes	0-10.800	0-17.280
Community of Interest	within the same group	43.200-86.400	69.120-138.240
	with the other group	0-10.800	0-17.280

Based on these data defining different network and traffic conditions, 42 experiments are performed on the heuristic design algorithm and its competitors. The

experiments, which are introduced in the following sections, are grouped by the number of nodes in the network because their results clearly show that the network size is the major condition effecting the performance of the heuristic design algorithm. To display the distributions of the solutions generated by the competitors as a statistical quality measure, Tests 4, 17, 26 and 39 are selected because together they span a subset of the tests which cover all possible network sizes, traffic types and traffic loads separately.

5.2.1. Tests on 8 Node Networks

The first series of tests are performed on small size networks. Having fixed the number of nodes by 8, 12 different example problems are generated by changing the density of the network, the type of the traffic and the load of the demand. Under each of these varying conditions, the heuristic design algorithm is compared to a total of 30000 results given by the idlest path routing algorithm and the random solution generators. The average and best results of the algorithms in these tests are shown in Table 5.10.

TABLE 5.10. Comparison of the Solutions on 8 Node Networks.

Test	Network Conditions			Heuristic Design Algorithm	Idlest Path Routing (10000 Runs)		10000 Feasible Random Solutions		10000 Random Solutions	
	# of Links	Traffic Type	Traffic Load	Maximum Link Load	Average Result	Best Result	Average Result	Best Result	Average Result	Best Result
1	27	uniform	light	**0.276**	0.330	0.276	0.413	0.276	0.364	0.276
2			heavy	0.542	0.641	**0.474**	0.768	0.505	0.774	0.511
3		1 center	light	0.363	0.378	**0.318**	0.461	0.320	0.499	0.328
4			heavy	0.670	0.725	**0.582**	0.843	0.610	1.008	0.630
5		2 groups	light	**0.387**	0.534	0.387	0.618	0.392	0.595	0.387
6			heavy	0.781	0.920	**0.721**	0.959	0.747	1.237	0.827
7	56	uniform	light	**0.141**	0.243	0.178	0.303	0.184	0.432	0.230
8			heavy	**0.332**	0.543	0.414	0.677	0.415	0.981	0.551
9		1 center	light	0.324	0.391	**0.270**	0.515	0.298	0.739	0.377
10			heavy	0.709	0.786	**0.581**	0.869	0.588	1.517	0.709
11		2 groups	light	0.304	0.408	**0.288**	0.538	0.330	0.754	0.375
12			heavy	**0.614**	0.870	0.620	0.935	0.691	1.625	0.807

As can be seen from Table 5.10, the results of the heuristic design algorithm are better than its competitors for Tests 1, 5, 7, 8 and 12. The algorithm performs well under uniform traffic, whereas the results for communities of interest type traffic are also

acceptable. But in the case where the traffic is centralized, the results of the algorithm are about 12 to 18 per cent worse than the best results, which belong to the idlest path routing algorithm. The heuristic design algorithm sorts the connection requests in decreasing order according to their peak rates, which means that, under centralized traffic conditions, the traffic to and from the center is routed first over the least utilized path. Then the remaining requests are handled. However, because of the nature of the traffic at the center, several links become congested before the optimization phase. Rerouting is also not as easy as with uniformly distributed traffic since the peak rate of the traffic from and to the center is too high to find an alternative route which has enough unallocated capacity. The idlest path routing algorithm, on the other hand, selects the connection requests to be routed at random. This way, it has the chance to find a better combination of assignments since the physical links are not overloaded at the vary beginning of the routing process.

Another important point to stress is that under heavily loaded communities of interest type traffic of Test 6, the idlest path routing algorithm and the feasible random solutions generator reject 2 per cent of the connection requests because of link capacity limitations, whereas the heuristic design algorithm manages to accept all requests without violating any constraints.

The idlest path routing algorithm outperforms the heuristic design algorithm in seven tests out of 12. The probability of producing better results in these seven tests varies between 0.09 and 29.63 per cent. The feasible random solutions generate better results in six tests giving probabilities between 0.06 and 4.25 per cent, whereas the random solutions give three better test results with probabilities changing from 0.05 to 1.51 per cent. Generally, the idlest path routing algorithm, which is a variation of the heuristic design algorithm itself, is the best competitor since it yields to better results than both of the random solutions algorithms. The distributions of the solutions generated in Test 4 and shown in Figures 5.5, 5.6 and 5.7 prove this aspect of the test algorithms more clearly. The result of the heuristic design algorithm in this particular test is 0.670, and its closest competitor, the idlest path routing algorithm produces 1769 solutions with results better than this.

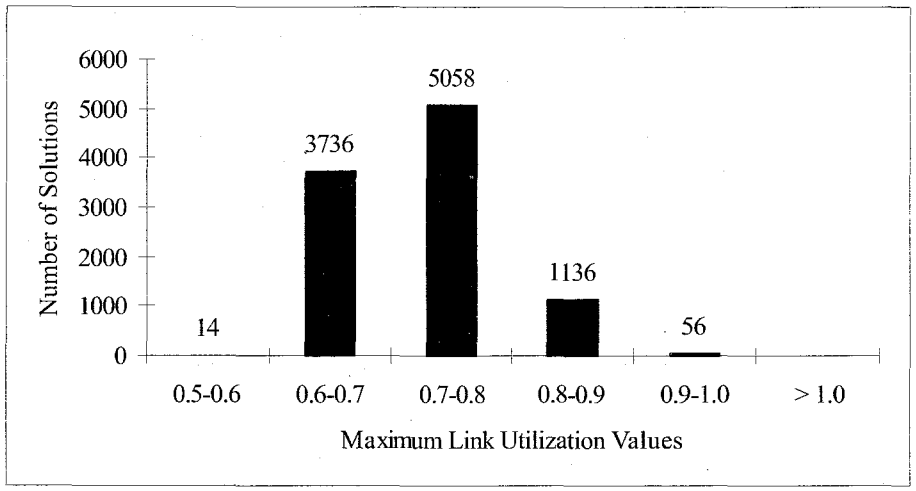

FIGURE 5.5. Distribution of 10000 Idlest Path Routing Solutions for Test 4.

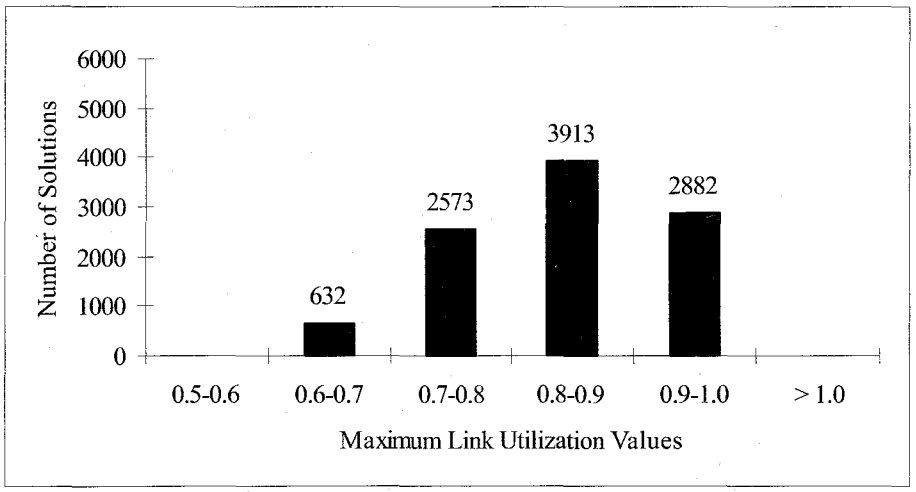

FIGURE 5.6. Distribution of 10000 Feasible Random Solutions for Test 4.

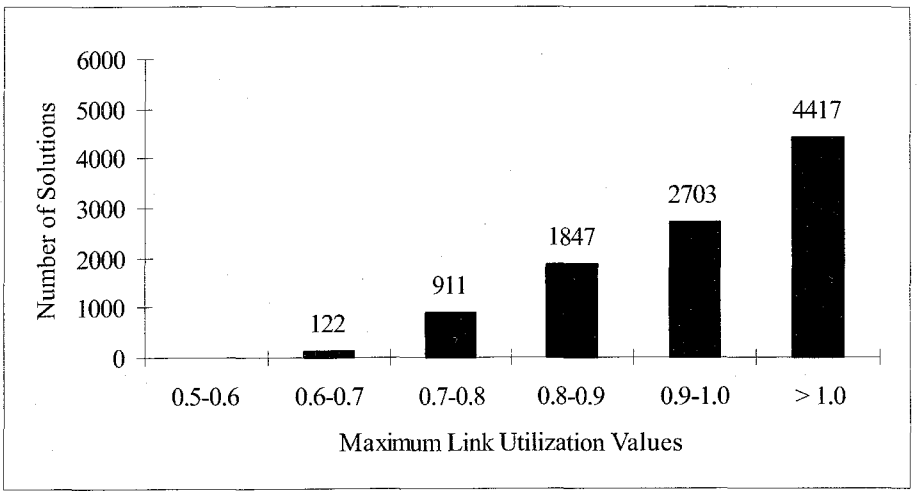

FIGURE 5.7. Distribution of 10000 Random Solutions for Test 4.

5.2.2. Tests on 16 Node Networks

The next test sequence is performed on networks of medium size with 16 nodes. Again, 12 different test conditions are generated by changing the density of the network, the type of the traffic and the load of the demand. Due to increasing time complexity of the test algorithms, however, the number of solutions generated by the idlest path routing algorithm as well as by the random solution generators is limited to 5000 for each of these tests. The average and best results of the algorithms in these tests are shown in Table 5.11.

TABLE 5.11. Comparison of the Solutions on 16 Node Networks.

Test	Network Conditions			Heuristic Design Algorithm	Idlest Path Routing (5000 Runs)		5000 Feasible Random Solutions		5000 Random Solutions	
	# of Links	Traffic Type	Traffic Load	Maximum Link Load	Average Result	Best Result	Average Result	Best Result	Average Result	Best Result
13	116	uniform	light	**0.261**	0.365	0.306	0.489	0.352	0.584	0.408
14			heavy	**0.520**	0.733	0.633	0.922	0.741	1.160	0.803
15		2 centers	light	**0.366**	0.439	0.372	0.628	0.413	0.826	0.502
16			heavy	0.752	0.893	**0.732**	0.974	0.842	1.650	0.982
17		2 groups	light	**0.424**	0.625	0.531	0.838	0.603	1.030	0.703
18			heavy	**0.874**	0.992	0.934	0.997	0.975	2.006	1.000
19	240	uniform	light	**0.189**	0.357	0.307	0.475	0.334	0.694	0.449
20			heavy	**0.365**	0.727	0.635	0.894	0.697	1.414	0.894
21		2 centers	light	**0.394**	0.477	0.404	0.758	0.473	1.107	0.615
22			heavy	0.821	0.942	**0.808**	0.982	0.893	2.238	1.000
23		2 groups	light	**0.382**	0.617	0.507	0.839	0.608	1.271	0.788
24			heavy	**0.694**	0.991	0.938	0.996	0.965	2.515	1.000

Table 5.11 shows the improvement of the performance of the heuristic design algorithm as the network grows larger. In 10 of the tests, it finds better results than all its competitors, and in the remaining two, where the traffic type is centralized again, there is only a slight difference of 2 to 3 per cent compared to the idlest path routing algorithm. In general, the former outperforms the latter by 7 to 35 per cent in communities of interest and 17 to 74 per cent under uniform traffic. The performance difference is even greater against random solutions, namely 35 to 91 per cent against 5000 feasible random solutions and 54 to 145 per cent against 5000 random solutions under uniform traffic conditions. The main reason behind this performance improvement is that the solution space is enlarged, and as a result of this, finding a good random solution becomes very hard. In fact, as explained later

in this chapter, this trend continues for larger networks and prove the advantage of a solution based on heuristic rules.

Similar to the case in the previous section, the idlest path algorithm and the feasible random solutions reject 5 and 3 per cent of the VCs, respectively, under heavily loaded communities of interest type traffic of Test 18 and Test 24 because of link capacity limitations while the heuristic design algorithm succeeds in routing all of the VCs. The competitors reject more connection requests as the network size increases, which can be seen in Chapter 5.2.3 more clearly. However, the heuristic design algorithm is not effected by the growth of the network. It can accommodate the increasing number of requests as a result of its principle of balancing the traffic load.

The idlest path routing algorithm is again a better competitor than both of the random solution generators against the heuristic design algorithm, as the distributions of the solutions generated in Test 17 show in Figures 5.8, 5.9 and 5.10. Still, its probability of producing better results than the proposed algorithm is only 0.02 per cent for heavily loaded centralized traffic and 0 elsewhere. In Test 17, the heuristic design algorithm offers a solution with a maximum link utilization of 0.424. When compared to the averages, this result is 47 per cent better than the idlest path routing algorithm, 98 per cent better than 5000 feasible random solutions and 143 per cent better than 5000 random solutions. In fact, the random solution generator produces 2622 infeasible solutions out of 5000, which is more than half of the total number. This is another indicator showing that as the network size grows, an approach based on certain rules is advantageous.

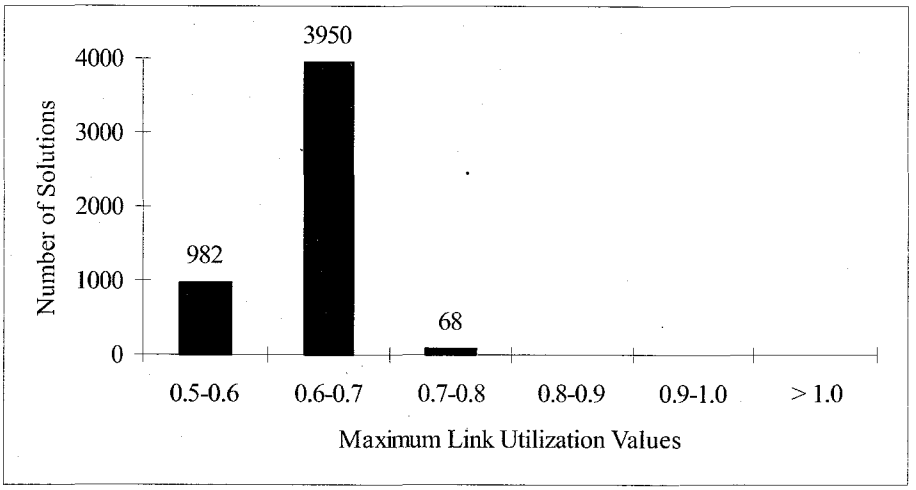

FIGURE 5.8. Distribution of 5000 Idlest Path Routing Solutions for Test 17.

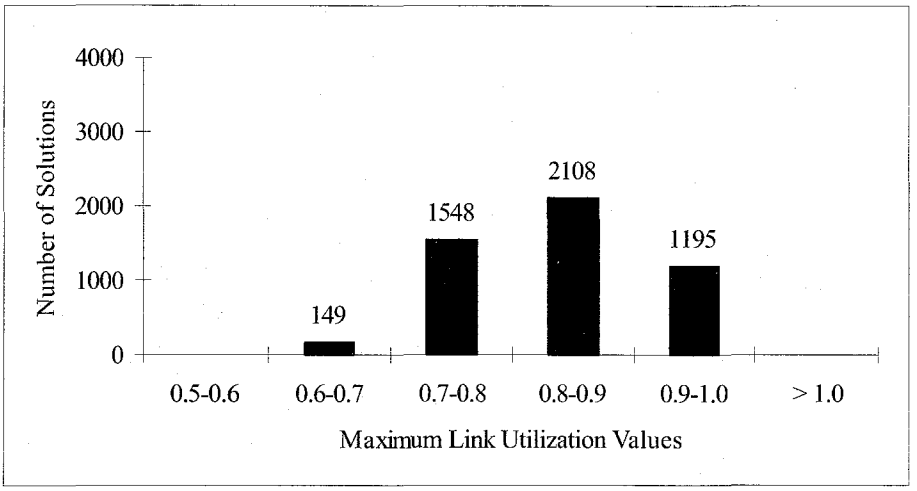

FIGURE 5.9. Distribution of 5000 Feasible Random Solutions for Test 17.

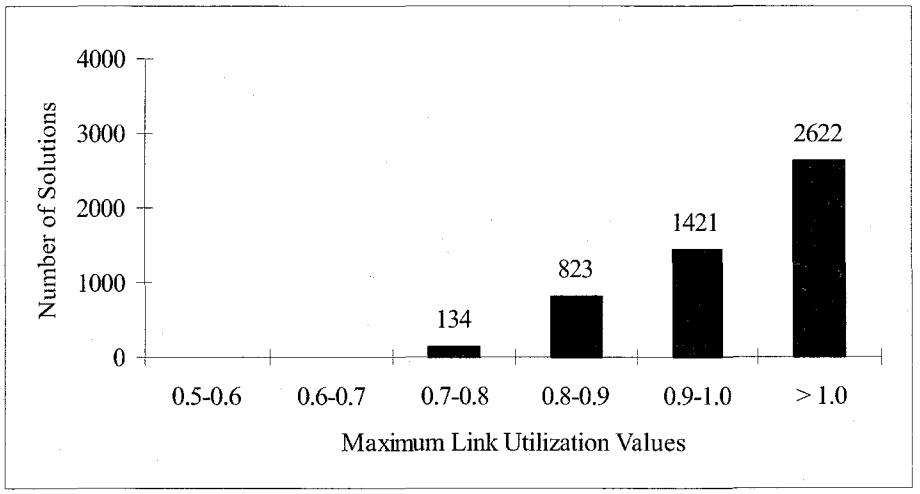

FIGURE 5.10. Distribution of 5000 Random Solutions for Test 17.

5.2.3. Tests on 32 and 64 Node Networks

Since the idlest path routing is the closest competitor to the heuristic design algorithm in the preceding comparisons, the remaining two series of tests, which are performed on large networks of 32 and 64 nodes, compare the performances of these two algorithms only to overcome the problem of excessive run times. In the first series involving 32 node networks, 5000 solutions are produced by the idlest path routing algorithm for each of the 12 tests. The results are shown in Table 5.12.

TABLE 5.12. Comparison of the Solutions on 32 Node Networks.

Test	Network Conditions			Heuristic Design Algorithm	Idlest Path Routing (5000 Runs)	
	# of Links	Traffic Type	Traffic Load	Maximum Link Load	Average Result	Best Result
25	486	uniform	light	**0.260**	0.435	0.398
26			heavy	**0.509**	0.887	0.826
27		4 centers	light	**0.391**	0.502	0.447
28			heavy	**0.773**	0.982	0.908
29		2 groups	light	**0.424**	0.718	0.657
30			heavy	**0.875**	0.999	0.988
31	992	uniform	light	**0.200**	0.453	0.417
32			heavy	**0.395**	0.919	0.850
33		4 centers	light	**0.410**	0.548	0.489
34			heavy	**0.816**	0.994	0.959
35		2 groups	light	**0.369**	0.761	0.695
36			heavy	**0.738**	0.999	0.988

Table 5.12 shows that the heuristic design algorithm has a better performance in all tests. When the results are observed in groups of traffic types, the differences are between 14 and 19 per cent for centralized traffic, between 13 and 88 per cent for communities of interest, and between 53 and 115 per cent for uniform traffic. Moreover, the idlest path routing algorithm rejects 12 per cent of the connection requests in Test 30 and Test 36 while the heuristic design algorithm rejects only 0.1 per cent in each test. The distribution of the solutions generated in Test 26 is given in Figure 5.11. In this test, the heuristic design algorithm offers a solution with a maximum link utilization of 0.509.

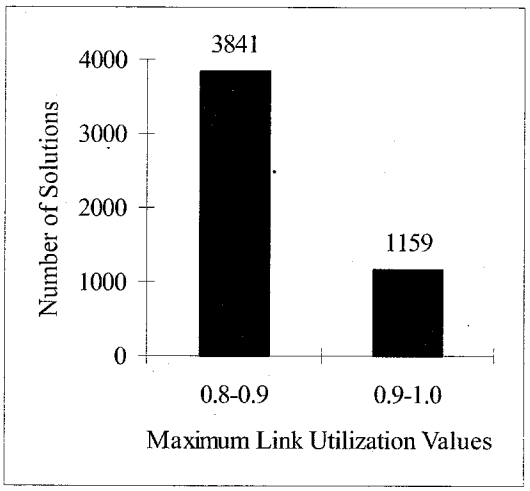

FIGURE 5.11. Distribution of 5000 Idlest Path Routing Solutions for Test 26.

The last group of tests concerns large networks of 64 nodes and consists of three individual tests with three different traffic types. Each of the tests offer 1000 solutions. The results are displayed in Table 5.13.

TABLE 5.13. Comparison of the Solutions on 64 Node Networks.

Test	Network Conditions			Heuristic Design Algorithm	Idlest Path Routing (1000 Runs)	
	# of Links	Traffic Type	Traffic Load	Maximum Link Load	Average Result	Best Result
37	1990	uniform	light	**0.285**	0.519	0.501
38		8 centers	light	**0.393**	0.549	0.515
39		2 groups	light	**0.495**	0.857	0.829
40	4032	uniform	light	**0.203**	0.551	0.531
41		8 centers	light	**0.419**	0.651	0.606
42		2 groups	light	**0.377**	0.937	0.907

The 64 node network tests show clearly that the trend of the heuristic design algorithm towards higher performance continues. In particular, it performs better than its competitor by 31 to 45 per cent for centralized traffic, 67 to 141 per cent for communities of interest and 76 to 162 per cent for uniform traffic, respectively. The distribution of the solutions generated in Test 39 is given in Figure 5.12. In this test, the heuristic design algorithm offers a solution with a maximum link utilization of 0.495.

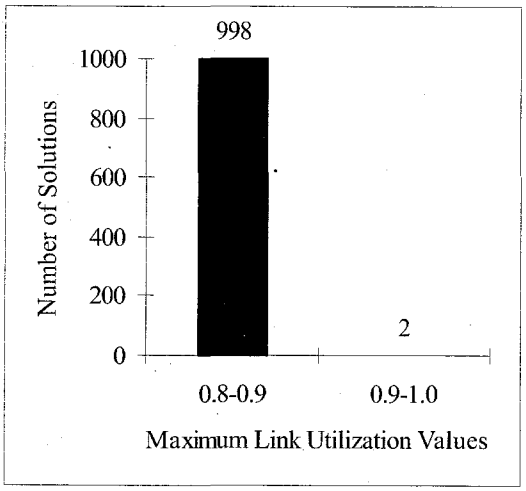

FIGURE 5.12. Distribution of 1000 Idlest Path Routing Solutions for Test 39.

5.3. Comments on the Algorithm Performance

In the preceding sections, the heuristic design algorithm developed in this study is compared to three competitors under different network and traffic conditions. The comparisons show that two major factors, namely network size and traffic type, greatly effect the performance of the proposed algorithm. In Figures 5.13 and 5.14, the heuristic design algorithm is compared to the idlest path routing algorithm, which proves to be the best of the competitors in the tests, to show their behaviors when these network evaluation criteria are changed.

The relation between the performances of the algorithms and the network size is shown in Figure 5.13, which is obtained by computing the average ratio of the difference between the results of both algorithms over the result of the heuristic design algorithm. In other words, the figure shows the factor by which the result of the heuristic design algorithm is better than the best result of the idlest path routing algorithm.

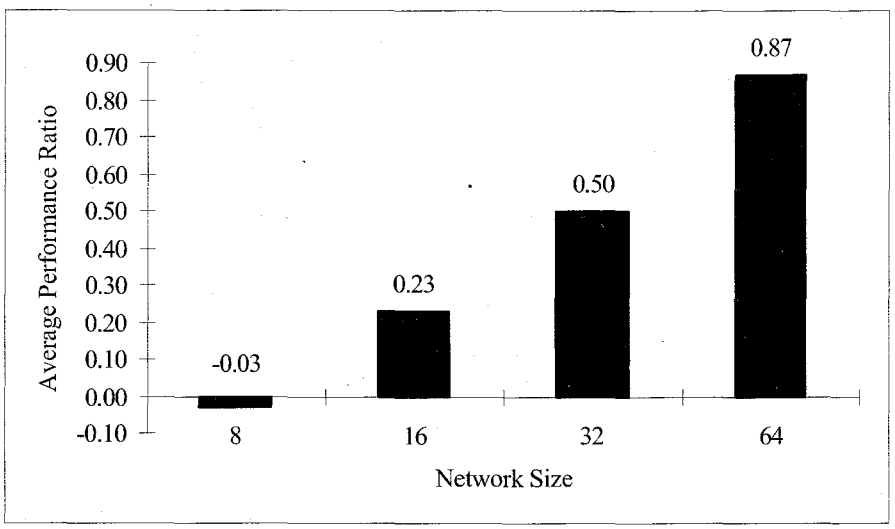

FIGURE 5.13. Quality of HDA Solutions as the Network Size Changes.

As previously mentioned, the idlest path routing algorithm has a better performance in small networks consisting of 8 nodes. The reason of this is that the solution space is not very large and a random design algorithm still has a chance to search it thoroughly and find a better solution than an algorithm with certain engineering rules. However, as the size of the network gets larger, the solution space grows and a heuristic algorithm which tries to design a VP network systematically has a greater chance of finding a better solution than a random one.

On the other hand, the relation between the algorithm performance and the traffic types can be explained by the distribution principles associated with these traffic patterns. Figure 5.14 is obtained in a way identical to Figure 5.13 and shows again the factor by which the heuristic design algorithm is better than its closest competitor.

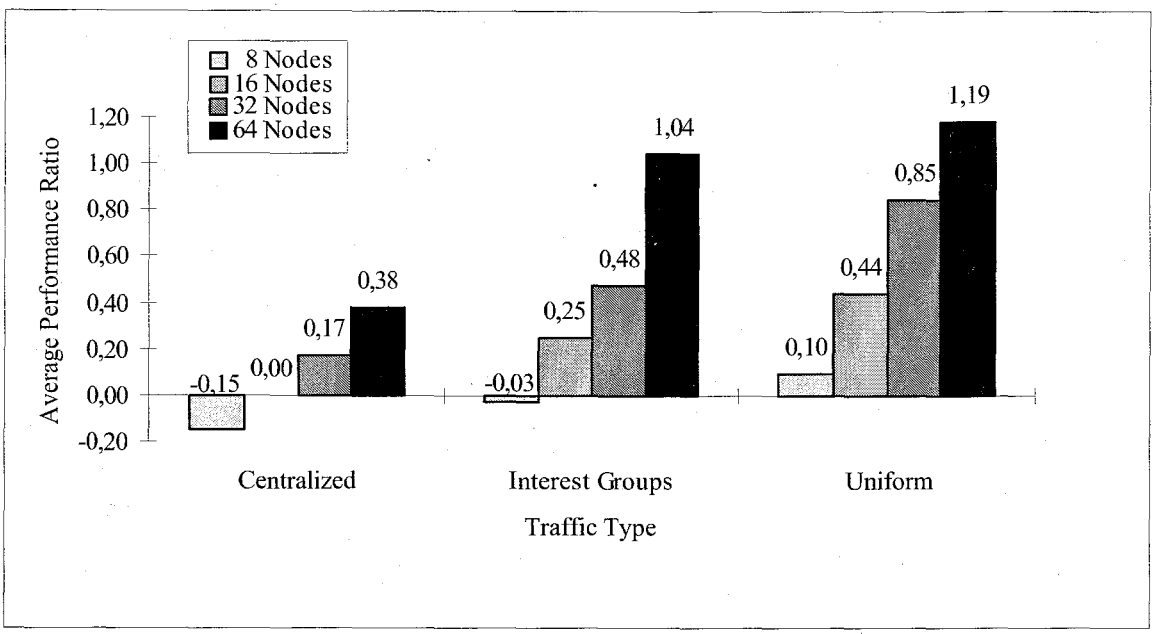

FIGURE 5.14. Quality of HDA Solutions as the Traffic Type Changes.

In fact, the trend of improved performance is observed by all traffic types as the number of the nodes in the network increases. Besides, the average of the results for these three kinds of traffic (10 per cent for centralized traffic, 43 per cent for groups of interest, and 64 per cent for uniform traffic) show that the solutions of the heuristic design algorithm is acceptable in all cases. However, these average values also show that the traffic type by itself has an effect on the algorithm performance. An explanation for the centralized traffic case is given in Chapter 5.2.1. In the case of communities of interest, the performance difference is much better since the distribution of the traffic load is uniform in each of the four blocks in the traffic matrix. This results in a somewhat more balanced utilization of links than the centralized traffic type, at least between the members of the same group. Finally, under uniform traffic conditions, the performance difference is the highest because of the homogenous nature of the traffic demand. This way, the possibility of rerouting of the VPs from highly utilized paths to less utilized ones is high enough to find a good traffic distribution since no links are overloaded too early, even before the optimization.

In general, the quality of the results given by the heuristic design algorithm on medium to large size networks is better than its competitors for communities of interest or uniform traffic, whereas it is acceptable for centralized traffic. For small size networks, the

solutions are not as good as the competitors, but they are still acceptable and can be applied much quicker since the algorithm finishes within a few iterations whereas its competitors need 10000 runs for better solutions. A slight decrease in the average number of VCs per VP is to be observed as the network size grows. However, the equivalent bandwidth concept still holds since it covers such cases, where a VP is assigned to only a few VCs, by its fluid-flow approximation. In the worst case, where a VP is assigned to just one VC, the equivalent bandwidth of the VP is very close to the peak rate of the VC. In fact, the stationary approximation can only be advantageous as a result of statistical multiplexing gain if several VCs share a VP.

As expected, the time complexity of the proposed design algorithm changes non-linearly with the size of the network. When running on a personal computer with a Pentium-100 microprocessor and 32 MB of RAM, the algorithm needs only a few seconds to finish for 8 node networks. For networks of 16 nodes, typical run times vary between 30 seconds and 3 minutes on the same test platform. In 32 and 64 node networks, additional termination conditions, like a maximum number of iterations or a minimum range of improvement, are needed if the algorithm is expected to finish in a particular amount of time, since the solution space is very large and, after a certain number of iterations, incremental changes are hard to find and not worth the time spent because they do not improve the maximum link utilization. Typical run times encountered in the experiments with 64 node networks change between 30 and 100 minutes on the test platform defined above. The growth of the heuristic design algorithm run times in accordance with the network size is shown in Figure 5.15. The values are normalized at 128 iterations.

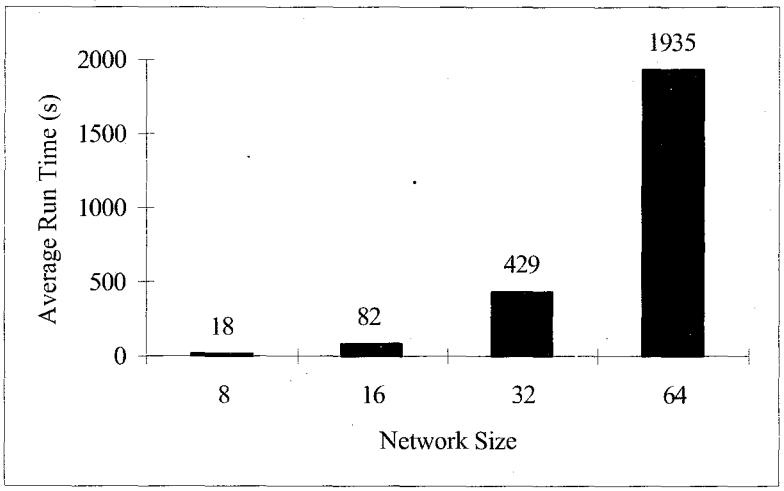

FIGURE 5.15. Average Run Times of HDA for 128 Iterations.

Another interesting point of view in the running time of the algorithm is the computational complexity. First of all, the key procedures explained in Chapters 4.3.2 and 4.3.3 are $O(N^2)$, where N is the number of nodes in the networks. Besides, in the worst case, the algorithm checks all links, their VPs and VCs during one iteration. So, the number of nodes and links are important factors which effect the duration of one iteration in the algorithm. On the other hand, the duration of the iterations can still not be predicted. An iteration might finish quickly if an improvement can be made on a more utilized link since these links are handled before others. This is the case for the first iterations of the optimization phase. However, iterations take longer times as the algorithm proceeds since improvements become rare and more links have to be checked to find one.

6. CONCLUSION

In this study, a method for designing the VP layout of an ATM network which makes possible the efficient use of the network resources under QoS constraints is proposed. The developed heuristic algorithm applies the equivalent bandwidth approach to compute the capacity requirements of the connection requests such that a desired QoS defined by the cell loss probability is guaranteed. The equivalent bandwidth concept is a computationally simple approximation to estimate capacity requirements, which also takes into account the effects of statistical multiplexing by using two approximations to calculate the equivalent bandwidth and taking the more beneficial result. The algorithm tries to minimize the maximum link utilization by applying VP and VC routing techniques under processing delay constraints, which are represented by a maximum allowable number of VP hops for the VCs. The quality of the solutions achieved by the heuristic design algorithm is compared to several competitors under varying network topologies and traffic conditions. The observations on the algorithm performance show that the developed method is able to facilitate an efficient use of network resources through the introduction of VPs.

Several issues have to be considered with the implementation of the proposed method in real-life networks. First of all, pure ATM networks are rare. In other words, today's networks mainly have a hybrid structure consisting of ATM switches and cross-connect nodes on the backbone, and older technologies like Ethernet at the level of individual users to protect the existing investment in commercial networks. In such networks, the proposed method has to be applied on the backbone, where the ATM cross connect nodes are the end-nodes of the network and the VPs are defined between them. Routing from the cross connect nodes to the actual user has to be done locally.

Another important implementation issue involves the handling of the case where new nodes or links are added to the backbone ATM network. In the current version of the algorithm, the whole VP network has to be redesigned to find a high quality solution where the newly added nodes or links are also taken into consideration. In small networks, where the algorithm can be used synchronously because of its speed, this is not a case of major

concern. But in medium to large networks, an extension to the algorithm is needed to make incremental changes in the VP routing scheme, concerning especially the new nodes, to offer a temporary solution to be applied until the heuristic design algorithm gives the new VP layout. The initialization phase of the proposed algorithm can find a temporary solution by creating VPs on the links of these new nodes. It should also be noted that, for small networks, the idlest path routing algorithm introduced as a competitor to the proposed heuristic design algorithm in the previous chapter can be embedded in the original algorithm to make use of its probabilistic nature.

Similarly, the failure of a node or link is a case where immediate action has to be taken. To handle the case of link failures, the heuristic design algorithm can be modified such that it applies the VP and VC movement techniques on the failed link to reroute its traffic. Since the time complexity of the algorithm comes from the process of looking for a link to improve its utilization and not from the search for an alternate route, a quick solution for the failed link can be achieved. Besides, the alternate routes do not have to be optimal to recover from link failures. In the case where a node fails, this procedure has to be repeated for every link connecting the failed node to the other nodes.

The equivalent bandwidth is an effective way to practically implement advanced network control functions because it provides a unified connection metric for network management. To further improve its accuracy, investigation of better approximations for β are necessary. The approach can also be used for satisfying call level QoS constraints like call blocking probability with assumptions of certain traffic conditions. Other issues concerning further work are reliability and recovery from failures. Secondary VPs can be created to backup every primary VP between two end nodes such that primary and secondary VPs are passed on completely disjoint physical paths to assure the network survivability.

Since the heuristic design algorithm developed in this study does not provide an optimal solution but a suboptimal one, the degree of its optimality is a subject to investigate as a further research topic.

REFERENCES

1. Plotkin, S., "Competitive Routing of Virtual Circuits in ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, pp. 1128-1136, August 1995.

2. Sato, K., S. Ohta and I. Tokizawa, "Broadband ATM Network Architecture Based on Virtual Paths," *IEEE Transactions on Communications*, Vol. 38, No. 8, pp. 1212-1222, August 1990.

3. Minzer, S. E., "Broadband ISDN and ATM," *IEEE Communications Magazine*, Vol. 27, No. 9, pp. 17-24, September 1989.

4. Ohta, S. and K. Sato, "Dynamic Bandwidth Control of the Virtual Path in an ATM Network," *IEEE Transactions on Communications*, Vol. 40, No. 7, pp. 1239-1247, July 1992.

5. Sato, K. and I. Tokizawa, "Broadband Transport Techniques Based on Virtual Paths," *Proceedings of Globecom '90*, pp.1269-1273, December 1990.

6. Chlamtac, I., A. Farago and T. Zhang, "Optimizing the System of Virtual Paths," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 6, pp.581-587 December 1994.

7. Burgin, J. and D. Dorman, "Broadband ISDN Resource Management: The Role of Virtual Paths," *IEEE Communications Magazine*, Vol. 29, No. 9, pp. 44-48, September 1991.

8. Sato, Y. and K. I. Sato, "Virtual Path and Link Capacity Design for ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 1, pp. 104-111, January 1991.

9. Gerstel, O., I. Cidon and S. Zaks, "The Layout of Virtual Paths in ATM Networks," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 6, pp. 873-884, December 1996.

10. Chlamtac, I., A. Farago and T. Zhang, "How to Establish and Utilize Virtual Paths in ATM Networks," *Proceedings of Globecom '93*, pp.1368-1372, December 1993.

11. Lin, F. Y. S. and K. T. Cheng, "Virtual Path Assignment and Virtual Circuit Routing in ATM Networks," *Proceedings of Globecom '93*, pp. 436-441, December 1993.

12. Aydemir, M. and Y. Viniotis, "Virtual Path Assignment in ATM Networks," *Proceedings of PCN'95*, pp. 382-400, 1995.

13. Shioda, S. and H. Uose, "VP Bandwidth Control Method for ATM Networks: Successive Modification Method," *IEICE Transactions on Communications*, Vol. E74-b, No. 12, pp. 4061-4068, December 1991.

14. Hou, C. J., "Routing Virtual Circuits with Timing Requirements in Virtual Path Based ATM Networks," *Proceedings of Infocom '96*, pp. 320-328, 1996.

15. Guerin, R., H. Ahmadi and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No 7, pp. 968-981, September 1991.

16. Ryu, B. H., H. Ohsaki, M. Murata and H. Miyahara, "Design Algorithm for Virtual Path Based ATM Networks," *IEICE Transactions on Communications*, Vol. E79-b, No. 2, pp. 97-107, February 1996.

17. Ryu, B. H., H. Ohsaki, M. Murata and H. Miyahara, "Design Method for Virtual Path Based ATM Networks with Multiple Traffic Classes, *Proceedings of ICCC'95*, pp. 336-341, August 1995.

18. Farago, A., S. Blaabjerg, W. Holender, B. Stavenow, T. Henk, L. Ast and S. Szekely, "Enhancing ATM Network Performance by Optimizing the Virtual Network Configuration," *Proceedings of PCN'95*, pp. 401-414, 1995.

19. Aneroussis, N. G. and A. A. Lazar, "Virtual Path Control for ATM Networks with Call Level Quality of Service Guarantees," *Proceedings of Infocom'96*, pp. 312-319, 1996.

20. Friesen, V. J., J. J. Harms and J. W. Wong, "Resource Management with Virtual Paths in ATM Networks," *IEEE Network*, pp. 10-20, September/October 1996.

21. Shioda, S., "Evaluating the Performance of VP Bandwidth Control in ATM Networks," *IEICE Transactions on Communications*, Vol. E77-b, No. 10, pp. 1175-1187, October 1994.

22. Logothetis, M. and S. Shioda, "Centralized VP Bandwidth Allocation Scheme for ATM Networks," *IEICE Transactions on Communications*, Vol. E75-b, No. 10, pp. 170-180, October 1992.

23. Hamchaoui, I. and F. Guillemin, "Resource Management in Virtual Private Networks over ATM," *Proceedings of ATM Workshop'97*, pp. 659-668, 1997.

24. Bertsekas, D. and R. Gallagher, *Data Networks*, Prentice-Hall, New Jersey, 1992.

25. De Prycker, M., *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Prentice-Hall, New Jersey, 1995.

26. Aoyama, T., I. Tokizawa and K. Sato, "Introduction Strategy and Technologies for ATM VP-Based Broadband Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 9, pp. 1434-1447, December 1992.

27. Aoyama, T., I. Tokizawa and K. Sato, "ATM VP-Based Broadband Networks for Multimedia Services," *IEEE Communications Magazine*, Vol. 31, No. 4, pp. 30-39, April 1993.

28. Shioda, S., H. Saito and H. Yokoi, "Sizing and Provisioning for Physical and Virtual Path Networks Using Self-Sizing Capability," *IEICE Transactions on Communications*, Vol. E80-b, No. 2, pp. 252-262, February 1997.

29. Awerbuch, B., Y. Azar, S. Plotkin and O. Waarts, "Competitive Routing of Virtual Circuits with Unknown Duration," *Proceedings of ACM-SIAM - the Fifth Symposium on Discrete Algorithms*, pp. 321-327, January 1994.

30. Rich, E. and K. Knight, *Artificial Intelligence*, McGraw-Hill, New York, 1991.

31. Kruse, R., *Data Structures and Program Design*, Prentice-Hall, New Jersey, 1987.