A virtual path routing algorithm for ATM networks based on the equivalent bandwidth concept

Bür, Kaan; Ersoy, Cem

Published in:
Computer Communications

DOI:
10.1016/S0140-3664(99)00160-7

2000

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Bür, K., & Ersoy, C. (2000). A virtual path routing algorithm for ATM networks based on the equivalent bandwidth concept. *Computer Communications*, *23*(4), 379-394. https://doi.org/10.1016/S0140-3664(99)00160-7

Total number of authors:
2

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

Kaan Bür burk@boun.edu.tr

Cem Ersoy ersoy@boun.edu.tr

Boğaziçi University
Computer Engineering Department
80815 Bebek İstanbul - Turkey

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 handle bursty traffic with unexpected fluctuations. The Asynchronous Transfer Mode (ATM) technology provides this flexibility by virtualizing network resources through the use of the virtual path (VP) concept. In this study, a method for designing a VP-based ATM network is proposed. The developed heuristic design algorithm applies VP routing and separation techniques to minimize the maximum link utilization under processing delay constraints. Each link is assigned a weight that reflects its current utilization. Using these weights, the VPs on highly utilized links are rerouted to less congested physical paths. The algorithm makes use of the equivalent bandwidth concept, which provides an efficient method to estimate capacity requirements of connection requests such that QoS requirements are met. The quality of the solutions achieved by the proposed method 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.

Keywords

Asynchronous Transfer Mode (ATM), equivalent bandwidth, network link utilization, virtual circuit (VC), virtual path (VP).

• This work is partially funded by State Planning Organization (DPT) under the grant number 96K120490.

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 with different quality of service (QoS) requirements have to be handled in the most efficient and economical way possible while traditional ones are kept maintained [1]. The network should also be able to evolve to meet unknown future demands for increased flexibility, capacity, and reliability [2]. Broadband Integrated Services Digital Networks (B-ISDN) support a wide range of applications with different QoS requirements in a flexible and cost-effective manner. The goal of B-ISDN is to define a user interface and network that meets varied requirements of these applications. The Asynchronous Transfer Mode (ATM) provides the required flexibility for supporting heterogeneous services in a B-ISDN environment [3].

Very large capacity fiber-optic transmission technologies have significantly reduced the transmission cost portion of the total network cost, making the node cost relatively high. This leads to the conclusion that the total network cost would be reduced most effectively by node cost reduction. 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. The fundamental advantage of this concept is that it allows the grouping of individual connections, also known as virtual circuits (VC), 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 % of processing time can be saved when VCs are routed on VPs rather than processed individually [4].

By reserving capacity on VPs, VC connections can be established quickly and simply. Transit nodes are free from the routing and bandwidth allocation procedures of call setup. This effectively reduces the processing loads of the transit nodes and leads to low cost node construction. However, reserving capacity in anticipation of expected traffic introduces the problem of decreased capacity sharing, i.e. a VP cannot exploit redundant capacity on another VP, and capacity segmentation occurs. Consequently, available transmission bandwidth is underutilized, the network throughput is decreased, and the total call blocking rate is increased [5]. Hence, the development of an efficient means of dynamically changing the VP bandwidth is essential to achieve high utilization of network sources. Adaptation to changes in traffic conditions can be provided by rearrangement of the bandwidth allocated to VPs [6]. When the connections on one VP increase, unused 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. The VP bandwidth can be set at the smallest value needed for containing the VCs. Capacity allocations and reallocations to VPs can be effectively handled with the help of the bandwidth data stored in the control processor of the end nodes and a central VP routing table containing VP paths and bandwidth allocations. The VP bandwidth can initially be set at the smallest value needed for containing the VCs. They can then be varied by merely modifying the bandwidth information at the control processor and the routing table without accessing the switches or the transit nodes. 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 [5]. However, the problem of determining the VPs in the network and their bandwidth allocation has not been satisfactorily addressed in the literature [6].

To identify the key characteristics of VP routing and capacity allocation, 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 [7].

(A) The number of VPs used for routing a VC (termed VP hop count) should be small in order to reduce the VC setup complexity introduced by bandwidth allocations and changes in the VC routing tables.
(B) The chosen route for a VC should be short in terms of the number of physical links it uses, or in terms of propagation delay, to efficiently utilize the communication network.
(C) The number of VCs handled by a VP should be small to keep the number of occupied entries in the VP routing tables (termed the load on the table) low.
(D) The number of VPs that share any link should be small so that if a link is disconnected, the number of VPs that need to be rerouted in order to by-pass the faulty link is small.

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 tries to distribute the network traffic as evenly as possible, since a balanced traffic load helps: (a) limiting the effect of link failures; (b) decreasing the chance of link saturation; (c) increasing the network robustness. As a result, the method can facilitate an efficient use of the network resources. This study is organized as follows: Chapter 2 provides a literature survey on VP network design goals and strategies, introduces the network model, and formulates the constrained optimization problem. In Chapter 3, the VP routing algorithm, its phases and key functions are described. Chapter 4 presents computational experiments, results, and comments on the algorithm performance. Finally, a summary, conclusions and subjects for further work are given in Chapter 5.

2. PROBLEM STATEMENT

The use of VPs improves the network performance in terms of call setup, cell-processing time, adaptability to variations in traffic, and administration. However, VP routing and capacity allocation issues have to be handled with great care in order to keep the negative effects of using VPs on capacity sharing, call blocking, processing load, and throughput at an acceptable level. An efficient method has to be developed so that a high quality VP layout design can be achieved. This study proposes a general model to exploit the benefits of the VP concept by trying to design a VP network that minimizes the selected cost function and meets the QoS requirements.

2.1. Review of Design Objectives

All VP routing and capacity allocation algorithms can be examined in two major categories, namely "synchronous" and "asynchronous". Synchronous algorithms update the VP capacity in real-time based on the observed demand for call establishment. The bandwidth of VPs can be expanded or contracted upon call arrival or departure. Asynchronous algorithms maintain a fixed VP distribution for a time period referred to as the update interval. The VP distribution and capacity allocation policy is computed at the beginning of the period and remains fixed during that period. Asynchronous algorithms can be further divided into two types, "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, which 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 are generally 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.

Algorithms can also be categorized by the objective (i.e. 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 one. On the other hand, 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.

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 links. It is assumed that G is strongly connected, i.e. there is a directed path between all node pairs [4, 8, 9, 10, 11]. A VP is defined by a starting node and an ending node, a directed route between these nodes, and a capacity assigned to this connection. The starting and ending nodes 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, which guarantees 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 [4, 8].

A possible objective is minimizing the maximum VP blocking rate. If there are delay limitations, a constraint can be included to keep the average number of VPs traversed from source to destination below a given constant. After each change made in the VP assignments, a capacity allocation algorithm adds bandwidth to VPs used by the worst blocked traffic streams in order to improve the VP blocking probability [9]. A cost function based on the VP blocking probability for selecting the VP that gets a capacity increase is employed [12].

Guaranteeing the timely delivery of messages is an important QoS requirement, 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 scheme that supports the timely and predictable delivery of messages. Given an ATM network, there are two issues to be considered. These are laying out VPs, and 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 satisfying the delay requirements for that VC, while not jeopardizing the QoS guarantees to other existing VCs.

In an attempt to design the VP-based ATM network, it can be required to consider a network topology and traffic pattern for minimizing a network construction cost while satisfying QoS requirements such as cell/call loss probabilities and cell delay times. 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 network cost is minimized by eliminating unnecessary links. The minimum required bandwidth of each 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 [10, 11].

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. 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 as a function of the weights of the links involved [1, 4, 8].

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. They are also referred to as successive modifications that can be seen as attempts at adapting to small traffic variations and minor network failures. To ensure responsiveness, the algorithms used should obtain results quickly, even at the expense of optimality, and 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 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 moderate network failures. They are also called global modifications. It is assumed that they occur much less frequently than successive modifications. Problems formulated in terms of global modifications typically attempt to optimize some global control objective, involving a more complex, time-consuming algorithm. Since changes to the VP topology are network-wide, the time required to implement the changes increases. 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. Finally, long term strategies may be employed to design a general VP overlay network, to be used at startup or after major network upgrades. They are likely based on knowledge of the physical network and possibly estimates of traffic patterns. The nature of long-term strategies imposes few constraints on the algorithm complexity, since solutions need not be found quickly [13].

Under the broad classification of these strategies, four basic VP management activities 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 capacity reallocation and topology reconfiguration in a network-wide scale.
(D) Long-term planning derives a static or general set of VPs and initial or minimum capacity assignments for them.

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 to trigger capacity increase on a set of VPs whenever a new connection cannot be accommodated [5]. 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. 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. The capacity allocation algorithm can be invoked when performance degrades beyond some operating state in terms of monitored entities like capacity reservation, capacity usage, and blocked traffic on each VP terminator. Capacity assignments can also be computed for each VP based on call blocking probabilities [12, 14]. Implementing a capacity reallocation involves changing bandwidth tables used by the call admission control (CAC) algorithm, as well as the parameters of any policing and shaping algorithm that may be applied per VP.

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 demands. 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 [13].

Global reconfiguration is useful when modifications are periodically made based on anticipated traffic conditions (e.g. hourly or daily [12, 14]). It is also useful in facilitating recovery from component failure. Global capacity reallocations can be made (without accompanying topology changes) when the observed call blocking probability exceeds some threshold [15]. The provision of multiple VPs to enhance connectivity 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 some of the main issues that have been addressed in the literature [5, 6, 10, 11, 12, 14, 15]. 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. Although major changes to the overlay network may take place on a time interval of weeks or months, these changes should be transparent to users in order to meet the requirements of the service providers and their customers [13].

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 such a scenario, the design of a general or static overlay network may be more concerned with limiting worst-case performance for any traffic pattern, and offering flexibility so that subsequent global or successive modifications (adapting to traffic variations) are easily made. 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 [13].

2.3. Mathematical Formulation of the Optimization Problem

The physical network topology with its nodes, links and link capacities provides the first set of input parameters of the VP routing problem. The other input is the traffic demand matrix in terms of capacities for all node pairs. Given these parameters, the proposed design model should provide the set of VPs with their routes, i.e. start, intermediate, and end nodes with 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. All nodes can be the source or the destination of the network traffic. The network links are 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. 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. Table 1 gives the notation used in this study.

Table 1. Mathematical Notation.

Symbol	Explanation
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_{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 bit 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)

The objective function to be minimized is the maximum link utilization of the network. This is an important aspect in the design and routing of VPs, since a homogenous distribution of the link utilization, together with 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 [12]. 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 others [16]. In this study, QoS is statistically handled by the application of the equivalent bandwidth concept, which is explained in Chapter 2.4. On the other hand, the minimization of the maximum link utilization helps the network meet the QoS requirements by providing it the necessary robustness and flexibility. Allowing more than one VPs with the same endpoints, the proposed model can cope with reliability issues [6] and also support multiple service classes. For the sake of simplicity, however, only one class of service is presented in the study. The objective function can be formulated as:

$$\min\left(\max\left(\frac{\sum_{P_{ij}} C'}{d_{ij}}\right)\right), \text{ for all } l_{ij} \in E \tag{2.1}$$

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 [4].

The cell loss probability is a 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 [17]. 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. 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 [10, 11].

One of the essential QoS requirements in ATM networks is the delay constraint on the delivery of cells. Since connection setup takes significantly more time than cell transmission, the processing of connection requests at intermediate nodes is a major delay factor. In other words, the network experiences larger delay times as the number of nodes taking part in the connection routing process increases. ATM networks facilitate the rapid movement of cell streams through the effective use of VPs, keeping intermediate node functions to a minimum [18]. Only VP switching nodes are involved in the connection setup process if connections are routed entirely on VPs. Therefore, the delay experienced by a VC can be limited by restricting the number of VPs traversed by that VC. In order to represent the delay restrictions on the connections, the maximum number of VPs that a connection is allowed to traverse is limited by the "VP hop count" parameter h. In this study, to reduce the complexity of the constrained optimization problem, connections are routed over paths consisting of at most 2 VPs, i.e. $h \leq 2$ [10, 11, 16, 19, 20]. Thus, VCs do not experience more than 1 VP switching process (and delay) on their routes. The choice of a maximum of 2 VP hops comes from a basic similarity with circuit-switched metropolitan networks such as telephone systems, suggesting that the knowledge and experience gained with the design and operation of the latter should be useful for ATM networks [21]. Once the model is implemented in its relatively simpler form, it can easily be improved to handle more general cases by modifying the route search procedure, which is explained in Chapter 3, to look for more than 1 pivot nodes for intermediate VP switching points. As a result, h can be determined according to network conditions such as Internet-like schemes, in which routes may use many VPs.

Because of the statistical multiplexing of connections and shared buffering points 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. 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 [22].

2.4. Equivalent Bandwidth

The equivalent bandwidth of a set of VCs multiplexed on a VP is defined as the amount of bandwidth required to statistically 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 [10, 11]. This model defines the general traffic structure of the network in terms of the peak rate and the duration of each connection, from which then the equivalent bandwidth necessary to statistically satisfy the QoS requirements is derived. 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 because they complement each other, capturing different aspects of the behavior of multiplexed connections, while remaining computationally simple [22].

In order to characterize the effective bit rate of a connection, a two-state fluid-flow model is adopted, where a source is either in an "idle state", transmitting at zero bit rate, or in a "burst state", transmitting at its peak rate. 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 [6]. 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 [22].

The equivalent capacity concept provides a unified metric for link utilization, 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 connection characteristics, existing network traffic, and desired QoS. Exact solutions are either intractable or, when available, incompatible with real-time requirements. Since the goal is to provide a simple while still reasonably accurate metric to measure and compare link utilizations, approximations must be used [22].

The first approximation, which is based on a fluid-flow model, accurately estimates the equivalent capacity when the impact of individual connection characteristics is critical. 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. In the case of a single two-state Markov source, the capacity to be allocated 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 buffer of finite size x, the equation satisfied by the equivalent bandwidth for ε (the desired QoS) is found to be of the form [22]:

$$\varepsilon = \beta \ \exp\left(-\frac{x\left(c'_k - \rho_k R_{peak,k}\right)}{b_k\left(1-\rho_k\right)\left(R_{peak,k} - c'_k\right)c'_k} \right) \tag{2.2}$$

The equivalent capacity for a single source can then be obtained by solving for c'_k in Equation 2.2. 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 2.2 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 [22]:

$$c'_k = \frac{\alpha b_k\left(1-\rho_k\right)R_{peak,k} - x + \sqrt{\left(\alpha b_k\left(1-\rho_k\right)R_{peak,k} - x\right)^2 + \left(4\alpha b_k \rho_k\left(1-\rho_k\right)R_{peak,k}\right)}}{2\alpha b_k\left(1-\rho_k\right)} \tag{2.3}$$

where $\alpha = ln\ (1/\varepsilon)$. Note that in the case of a continuous bit stream connection $\rho_k = 1$ and $b_k = \infty$, and taking limits in Equation 2.3 yields the expected result $c'_k = R_{peak,k}$.

In the case of multiple superposed sources, the value of the equivalent capacity $C'(F)$ given by the flow approximation for n multiplexed connections is defined by [22]:

$$C'(F) = \sum_{k=1}^{n} c'_k \tag{2.4}$$

where c'_k values are determined from Equation 2.3.

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. 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 capacities much larger than their mean bit rates are multiplexed. This is essentially the case for connections with long burst periods and relatively low utilization. It is then reasonable to allocate enough bandwidth to make the probability of an overload condition equal to the desired buffer overflow probability. 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, which is given by [22]:

$$C'(S) \cong m + \alpha' \sigma \tag{2.5}$$

with

$$\alpha' = \sqrt{-2\ln(\varepsilon) - \ln(2\pi)}\,,\ m = \sum_{k=1}^{n} m_k\,,\ \text{and}\ \sigma^2 = \sum_{k=1}^{n} \sigma_k^{\ 2} \tag{2.6}$$

where m is the mean and σ is the standard deviation of the aggregate bit rate.

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)* [22]. That is:

$$C' = \min\left\{C'(F), C'(S)\right\} \tag{2.7}$$

In summary, the problem can be formulated as follows.

Given:	G, the physical network topology with nodes, links and link capacities;
	C, the set of all VCs 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 $\leq \varepsilon$;
	Number 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.

3. HEURISTIC DESIGN ALGORITHM

In order to solve the complex optimization problem described above in a reasonable amount of time, the design algorithm is based on heuristics. It is a search algorithm looking for a high quality solution in the domain of valid VP assignments. The algorithm consists of initialization and optimization phases. In the former, a starting point is found which is a feasible 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 algorithm is given in Figure 1.

```
Program: VP Design

Phase 1: Initialization
Compute equivalent bandwidths of all VCs;
Create VPs for node pairs connected by direct links;
Try to route all VCs over these VPs;
Phase 2: Optimization
Repeat
        Sort remaining VCs in descending order of bandwidth;
        For every VC in the list try to find the most idle alternate route;
        Sort the physical links in descending order of utilization;
        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 a better alternate route;
                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 a better alternate route;
                Until a VC is rerouted successfully or end of VC list;
Until no improvement can be achieved on any link;
```

Figure 1. Pseudo-Code for the Heuristic Design Algorithm (HDA).

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.

3.1. Initialization Phase

In this phase, a set of VPs, P, is initialized by creating a VP p_{ij} on every physical link l_{ij} of the network under the following conditions:

$$t_{ij} > 0 \text{, for all } l_{ij} \in E \tag{3.1}$$

$$d_{ij} \geq t_{ij} \text{, for all } l_{ij} \in E \tag{3.2}$$

The inequalities given in Equation 3.1 and Equation 3.2 imply that there is sufficient physical link capacity to accommodate the traffic demand between the nodes i and j. Every p_{ij} 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 selection order of the VPs is irrelevant since there is no multiplexing of VCs on VPs or physical links yet. This initial VP layout where every physical link has a VP makes use of direct physical routes.

After this first step, the remaining VCs, i.e. the ones representing 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 and / or new VPs, i.e. the idlest one in terms of link utilization, is sought after. At the end of the initialization phase, 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.

3.2. 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. 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. These reallocations may lead to free bandwidth that can be used for reallocating VPs or VCs on the worst loaded link in later turns.

In the optimization phase, the possibility of moving a VP from P_{ij}, the set of the VPs using l_{ij}, to other links is checked for every physical link l_{ij}. The aim of changing the physical route of a VP is the reduction of the link utilization u_{ij} of l_{ij}. The VP to be rerouted first is the one with the largest capacity. 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. 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, 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. This procedure is repeated this way until all VCs from C_{ij} are checked.

The VP movement activity is prior to the VC movement activity because the statistical multiplexing gain implies that the total required bandwidth is increased when a single VP is separated into several VCs. On the other hand, the statistical multiplexing gain is not lost when a whole VP is rerouted. Separation of one or more VCs from a VP also requires equivalent bandwidth recalculations on the old and new routes, whereas there is no such need for VP movements. In the case where more than one of the alternate routes offer the same amount of improvement, use of existing VPs is always encouraged since it increases capacity sharing and decreases call blocking probabilities. 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 whole process is repeated. The optimization phase is terminated when an iteration is completed without an improvement for any link.

3.3. Key Procedures

VP and VC movement activities mentioned in the previous section make use of several heuristic methods to achieve a satisfactory VP layout in terms of QoS. Three of these methods are essential for the developed algorithm. They are the procedures responsible for calculating the equivalent bandwidth values, searching for the existing or new route alternatives, and finding the idlest, i.e. least congested, physical path among these routes. In the following sections, these key procedures are explained in more detail.

3.3.1. 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 2.3. 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 QoS constraints on the connections. 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 2.4, 2.5, and 2.6. The computational simplicity of performing these calculations gives the algorithm the chance to make them repetitively when trying various "what if" cases and checking different possibilities.

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.

3.3.2. Search for the Best Alternate 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 that 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 set of valid solutions and choose the best candidate among them.

In any case, the procedure first examines P, the set of VPs, 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, 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 the next section.

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, the use of existing VPs is always encouraged since it increases capacity sharing and therefore decreases call blocking probabilities.

3.3.3. Idlest Path Technique

The purpose of this procedure is to find the idlest physical route between a given source-destination pair and return the list of nodes as well as the maximum congestion on the path found. 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 the source and the destination nodes given. 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 as follows. 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 that do not have actual physical links, the procedure represents the fact 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 and adds it to the tree. It updates the "lengths" of the connections to nodes not yet reached as follows. 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 idlest 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 1. The pseudo-code for the procedure is given in Figure 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 2. Pseudo-Code for the Idlest Path Procedure.

4. 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 achieved by using the same objective function and set of constraints provided in the literature to compare the quality of the proposed algorithm directly. 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, competitor algorithms are developed in order to evaluate the quality of the heuristic design algorithm results. The first competitor, which is called "Idlest Path Routing", is a variation of the proposed heuristic design algorithm itself, where the initialization phase and the sorting of waiting VCs in the original algorithm are omitted. The solutions found by the heuristic design algorithm are also evaluated using statistical quality measures implemented as two additional competitors. These random search algorithms route arbitrarily chosen VCs on randomly selected combinations of physical links, disregarding all constraints concerning QoS like cell loss probability and delay. The idea is to see the performance of the heuristic design algorithm based on the distribution of the competitor results in the solution space and have a statistical notion about the quality of the heuristic solutions.

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.

4.1. Demonstrative Example

The purpose of this section 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. Two types of physical links according to their capacities are used. These are STS-3 and STS-12 links defined as 155.520 Mbps and 622.080 Mbps, respectively. The offered traffic load in the network is shown in Table 2, which is obtained by scaling the results of a metropolitan area network simulation [10, 11, 12] such that the capacity limitations do not become a bottleneck for the demonstration. Table 2 shows the R_{peak} values of the traffic demands.

Table 2. 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	20.980
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 $x=5$ *Mb*, $\varepsilon=10^{-5}$, $h=2$, $\rho_i=0.5$ and $b_i=100$ *ms*. The algorithm creates a total of 45 VPs, which are listed in Table 3, and routes all 56 VCs over these VPs.

Table 3. VPs of the Network (bandwidths are in Mbps).

VP #	VP Route (Nodes)	VP Bandwidth	VP #	VP Route (Nodes)	VP Bandwidth	VP #	VP Route (Nodes)	VP Bandwidth
1	1-3	298.810	**16**	5-4	117.170	**31**	8-5	62.910
2	1-4	123.610	**17**	5-6	49.420	**32**	8-4-3	186.330
3	1-6	199.920	**18**	5-7	96.750	**33**	3-7-5-1	100.880
4	1-7	57.090	**19**	5-8	24.190	**34**	3-6-8-5-1	75.140
5	1-8	19.720	**20**	6-1	20.900	**35**	6-8-5	61.860
6	2-1	295.090	**21**	6-2	136.490	**36**	4-5-7-1	71.010
7	3-7-5-4-2	121.940	**22**	6-7	60.480	**37**	6-1-4-5	48.820
8	3-5	75.780	**23**	6-8	175.070	**38**	8-5-7	26.000
9	3-6	171.290	**24**	7-1	155.690	**39**	5-7-8-2	0.000
10	3-7	100.120	**25**	7-4	39.550	**40**	8-5-1	20.560
11	4-2	22.640	**26**	7-4-5	17.320	**41**	1-7-5	31.850
12	4-3	100.640	**27**	7-6	51.900	**42**	2-1-8	20.980
13	4-5	88.750	**28**	7-8	41.100	**43**	5-1-4-2	20.540
14	5-1	15.640	**29**	8-2	20.400	**44**	7-1-6-2	22.320
15	5-3	76.500	**30**	8-4	13.480	**45**	1-4-5-8	26.540

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 path. The assignments of VPs to VCs are shown in Table 4. 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 76 % of the assignments is 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 4. VP - VC Assignments.

s \ d	1	2	3	4	5	6	7	8
1		3-21	1	4-25	41	3	4	45
2	6		6-1	6-2	6-41	6-3	6-4	6-5
3	34	7		33-2	8	9	10	9-23
4	36	11	12		13	36-3	13-18	13-19
5	14	43	15	16		17	18	19
6	20	21	23-32	35-16	37		22	21-42
7	24	44	24-1	25	26	27		28
8	38-24	29	32	30	31	40-3	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 maximum utilization of 0.519 is experienced on the link from node 3 to node 7, which is traversed by 3 VPs. The average congestion is 0.354. The resulting link utilizations are shown in Table 5.

Table 5. Physical Link Utilizations after the VP Routing.

s \ d	1	2	3	4	5	6	7	8
1			0.480	0.353		0.357	0.143	0.262
2	0.508							
3		0.000			0.487	0.396	0.519	
4		0.265	0.461		0.406			
5	0.374		0.492	0.384		0.318	0.311	0.326
6	0.448	0.255					0.389	0.502
7	0.400			0.366	0.409	0.334		0.264
8		0.033		0.321	0.396			

The algorithm is demonstrated for a small size network. In what follows, comparisons are made with the competitors to show the quality of the results. Table 6 displays the maximum link utilization results of the random solutions and the idlest path routing solution in comparison with the result of the heuristic design algorithm. The "Best Result" row shows that the heuristic design algorithm finds a lower value than all its competitors for the maximum link utilization. The result achieved by the heuristic design algorithm is 1.54 % better than the best result of 10000 idlest path routing solutions. The performance difference grows when the heuristic design algorithm is compared to the random solution generators.

Table 6. Comparison of the Maximum Link Utilizations for the Demonstrative Example.

	Heuristic Design Algorithm	**Idlest Path Routing (10000 Runs)**	**10000 Feasible Random Solutions**	**10000 Random Solutions**
Best Result	0.519	0.527	0.584	0.823
Average Result	0.519	0.710	0.896	1.985
Worst Result	0.519	1.000	1.000	4.023

Another advantage of the heuristic design algorithm in this case is that it can be used for obtaining VP assignments in real time because of its speed. Since the use of heuristics prevent infeasible moves in the solution space of the 8-node network, it takes the proposed algorithm a few (namely, 13 for the example) iterations to find the result. The design algorithm quickly makes incremental changes as necessary in the initial VP layout and improves the objective function until it reaches a steady state. On the other hand, the competitors have to make 10000 consecutive trials to find a good solution and provide a reliable statistical quality measure. Figure 3 displays the distribution of the competitor solutions for the demonstrative example, where the heuristic design algorithm fits into the leftmost part of the graph since it achieves a maximum link utilization of 0.519.

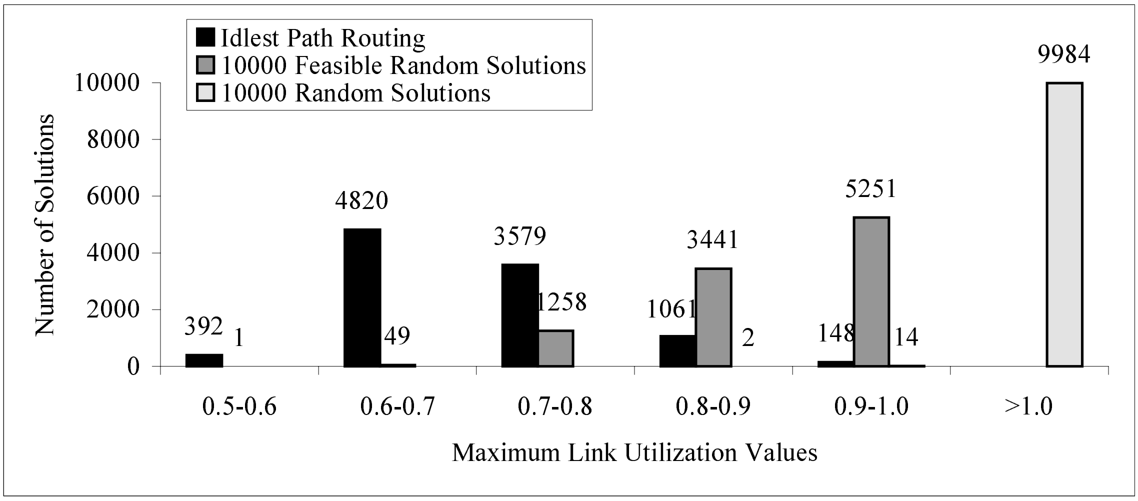


Figure 3. Distribution of the Competitor Solutions.

The quality of the competitor results improves as intelligence replaces randomness. Figure 3 shows how the demonstration results shift from larger utilization values found by the random solutions towards smaller values achieved by idlest path routing. In other words, idlest path routing solutions are distributed among smaller utilization values than the solutions of the statistical quality measures. Therefore, in the next section, only those results are presented which compare the heuristic design algorithm with its closest competitor, idlest path routing.

4.2. Quality of Results

Two test criteria out of the four mentioned at the beginning of this chapter greatly effect the performance of the proposed algorithm. These are the network size and the traffic type. In the experiments, four different network sizes with 8, 16, 32 and 64 nodes are used. Within these networks, three different traffic types are simulated. The first traffic type is uniform, where the demand between nodes is uniformly 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. For this traffic type, every 8^{th} node is defined as a new center in the network, making the total number of centers 1, 2, 4 and 8 for 8, 16, 32 and 64 nodes, respectively. 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 when compared to the traffic between members from different groups. In all of the four different cases regarding the network size, nodes build two interest groups of equal size, resulting in groups of 4, 8, 16 and 32 in 8, 16, 32 and 64 nodes, respectively.

In Figure 4 and Figure 5, 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. In each of these figures, a positive value shows that the heuristic design algorithm outperforms the idlest path routing algorithm by a factor given by that normalized value, and a negative value means the opposite. The relation between the algorithm performance and the network size is shown in Figure 4, which is obtained as follows. First, the tests are grouped according to network size. Then, for each test within each group, the ratio of the difference between the maximum link utilizations of both algorithms over the maximum link utilization of the heuristic design algorithm is computed. Finally, the average of these ratios is calculated for each of the four groups. In other words, Figure 4 shows the factor by which the maximum link utilization of the heuristic design algorithm is better than the minimum, meaning the best, maximum link utilization found by the idlest path routing algorithm.

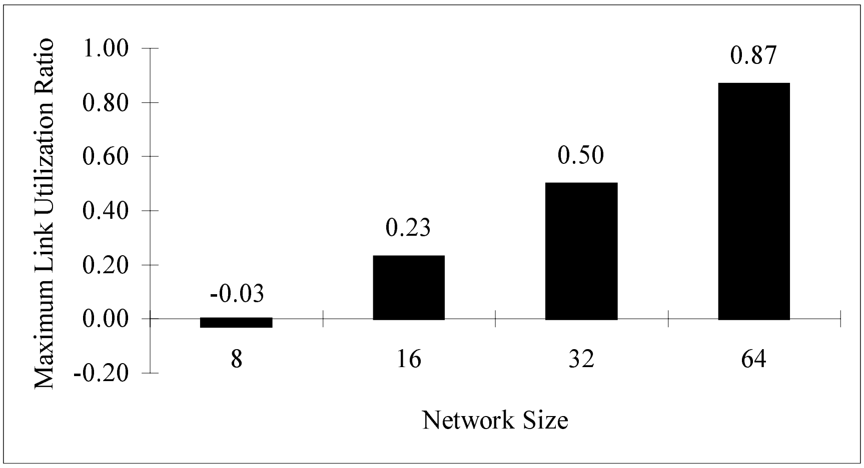


Figure 4. Quality of HDA Solutions as the Network Size Changes.

Figure 4 shows that the idlest path routing algorithm has a slightly better performance in small networks consisting of 8 nodes. This is mainly because the solution space is small, giving a design algorithm with randomness a chance to search it thoroughly and find a better solution than an algorithm with certain engineering rules. It should be noted, however, that this chance is less than 3 %. On the other hand, the heuristic design algorithm is much faster than idlest path routing, since it finds a good enough solution without making 10000 trials. In addition, idlest path routing does not guarantee a better solution than that of the heuristic design algorithm in all cases since it contains random elements, whereas the heuristic design algorithm provides a high quality solution in all cases. Finally, as the network size gets larger, the solution space grows and a heuristic algorithm that tries to design a VP network systematically has a greater chance of finding a better solution than a random one.

The relation between the algorithm performance and the traffic type can be explained by the distribution principles associated with these traffic patterns. Figure 5 is obtained in a way similar to Figure 4 and shows again the factor by which the heuristic design algorithm is better than its closest competitor. For the traffic types used in this study, a trend towards improved performance is observed as the number of nodes in the network is increased. In addition, the maximum link utilization ratios point out that the heuristic design algorithm provides acceptable solutions for networks of variable size under these traffic conditions. On the other hand, the results also express that the traffic type has an impact on the algorithm performance. Specifically, the average of the results is 10 % for centralized traffic, 43 % for groups of interest, and 64 % for uniform traffic, showing that the performance of the heuristic design algorithm generally improves as the traffic imbalances fade. The algorithm behavior cannot be precisely predicted from these results for the traffic conditions not considered in this study. However, it can be argued that for the traffic types, which can be modeled as combinations of the three simulated in this study, similar results are to be expected.

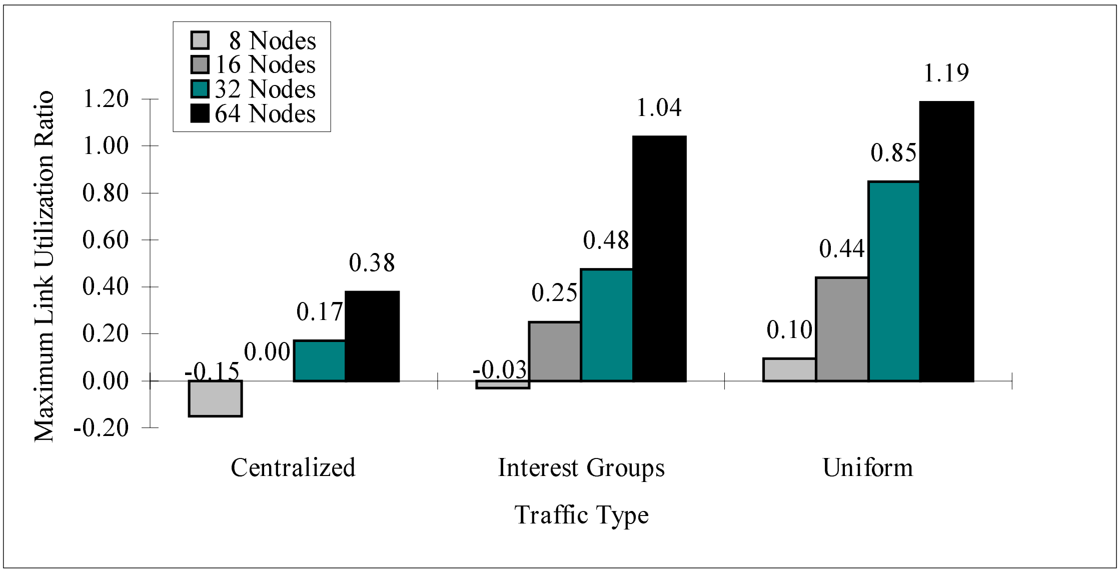


Figure 5. Quality of HDA Solutions as the Traffic Type Changes.

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 a chance to find a better combination of assignments since the physical links are not overloaded at the very beginning of the routing process. 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.

Another point to mention is the computational complexity of the heuristic design algorithm. First of all, the key procedures are of $O(N^2)$, where N is the number of nodes in the network. Besides, in worst case iterations, the algorithm checks all links, their VPs and VCs. So, the number of nodes and links are important factors that effect the duration of iterations in the algorithm. However, the duration of the iterations can still not be predicted. Iterations 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 as the algorithm proceeds since improvements become rare and more links have to be checked to find one.

5. CONCLUSION

In this study, a method for designing the VP layout of an ATM network, which makes possible the efficient use of the available bandwidth under QoS constraints, is proposed. The heuristic design algorithm applies the equivalent bandwidth approach to compute the capacity requirements of the connections such that a QoS defined by the cell loss probability is guaranteed. 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. Rerouting of a VP or a VC is decided on by comparing its current utilization with that of alternate routes. The utilization of a path is computed with the aid of "weights" assigned to each link to represent its congestion. 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.

According to Private Network-to-Network Interface (P-NNI) Phase 1 protocol defined by the ATM Forum, the VC routing scheme must ensure that a connection request is routed along a path that leads to the destination and has a high probability of meeting the QoS requested in the connection setup, that is, of traversing switches whose local CAC will not reject the call. In addition, the P-NNI protocol supports routing schemes where the initial node of a connection request determines the entire route to the final destination, since in these schemes it is relatively easy to decide on the path based on the requested QoS and the knowledge of the network state [23]. Thus, the VP/VC routing model proposed in this study is supported by P-NNI, since the heuristic design algorithm meets both of these necessary conditions.

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 these types of 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 can be done locally. In terms of further work, the proposed algorithm can be implemented and tested in such a real network.

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 taken into consideration. In small networks, where the algorithm can be used in real time environments as a result of its speed, this is not a case of major concern. However, 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.

Different applications require different types of connections in ATM networks. In general, these are point-to-point, point-to-multipoint, and multipoint-to-multipoint connections. In terms of connection types, this study considers only point-to-point connections, where the connection is established between exactly two entities. Most current services fall into this category. However, multimedia applications are of increasing importance. Various multimedia applications involve a group of users and require point-to-multipoint or multipoint-to-multipoint connections, such as video distribution and video conferencing. Some challenges to providing multimedia services in ATM are multiple call setup, synchronization of users, multipoint cell transfer, and synchronization of integrated applications. Managing the communication between a group of users is much more complex than point-to-point communications. In addition, the extension of existing point-to-point application models to characterize the integrated environment of a multimedia service is not a trivial task and has not yet been satisfactorily addressed in the literature. Therefore, point-to-multipoint and multipoint-to-multipoint routing can be considered as future research issues for this study.

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 guarantee an optimal solution, the degree of its optimality is a subject to investigate as another further research topic.

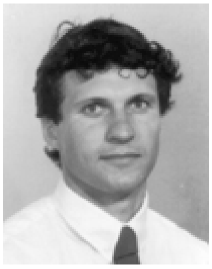
REFERENCES

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. 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.
5. 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.
6. Onvural, R. O., *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House Inc., Norwood, 1995.
7. 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.
8. 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.
9. Aydemir, M. and Y. Viniotis, "Virtual Path Assignment in ATM Networks," *Proceedings of PCN'95*, pp. 382-400, 1995.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. Tham, Y. K. and A. Arvidsson, "Virtual Path Routing for ATM Networks," *Proceedings of the Fifth IFIP International Workshop on the Performance Modelling and Evaluation of ATM Networks*, pp. 60/1-60/5, 1997.
19. 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.

20. 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.
21. Krishnan, K. R. and R. H. Cardwell, “Routing and Virtual-Path Design in ATM Networks,” *Proceedings of Globecom’94*, pp. 765-769, November 1994.
22. 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.
23. Alles, A., “ATM Internetworking,” *Proceedings of the Engineering InterOp’95*, March 1995.

VITAE

Kaan Bür received his BS degree in control and computer engineering from İstanbul Technical University in 1995 and his MS degree in computer engineering from Boğaziçi University in 1998. Currently, he is a PhD candidate in the Computer Engineering Department of Boğaziçi University. His research interests include high-speed networking, ATM networks, wireless and multimedia communications.



Cem Ersoy received his BS and MS degrees in electrical engineering from Boğaziçi University in 1984 and 1986, respectively. He received his PhD in electrical engineering from Polytechnic University in 1992. Currently, he is an associate professor in the Computer Engineering Department of Boğaziçi University. His research interests include performance evaluation and topological design of communication networks, wireless and multimedia communications.