Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-based Dynamically Reconfigurable Wide-Area ATM Networks

D. Medhi
Computer Science Telecommunications Program
University of Missouri-Kansas City
5100 Rockhill Road
Kansas City, MO 64110 USA
Tel: +1 816 235-2006
E-mail: dmedhi@cstp.umkc.edu
Fax: +1 816 235-5159

August 1994
Revised April 1995

Abstract

Virtual Path (VP) concept has been gaining attention in terms of effective deployment of ATM (Asynchronous Transfer Mode) networks in recent years. In a recent paper, we have outlined a framework and models for network design and management of dynamically reconfigurable ATM networks based on the virtual path concept from a network planning and management perspective. Our approach has been based on statistical multiplexing of traffic within a traffic class by using a virtual path for the class and deterministic multiplexing of different virtual paths, and on providing dynamic bandwidth and reconfigurability through virtual path concept depending on traffic load during the course of the day. In this paper, we discuss in detail a multi-hour, multi-traffic class network (capacity) design model for providing specified quality-of-service in such dynamically reconfigurable networks; this is done based on the observation that statistical multiplexing of virtual circuits for a traffic class in a virtual path, and the deterministic multiplexing of different virtual path lead to decoupling of the network dimensioning problem into the bandwidth estimation problem *and* the combined virtual path routing and capacity design problem. We discuss how bandwidth estimation can be done, and then how the design problem can be solved by a decomposition algorithm by looking at the dual problem and using subgradient optimization. We provide computational results for realistic network traffic data to show the effectiveness of our approach. We show that, for the test problems considered, our approach does between 6 % to 20 % better than a local shortest path heuristic. We also show that considering network dynamism through variation of traffic during the course of a day by doing dynamic bandwidth and virtual path reconfiguration can save between 10 to 14 % in network design costs compared to a static network based on maximum busy hour traffic.

Keywords: wide-area ATM networks, dynamic virtual path routing, multi-hour network capacity design, optimization model, duality and subgradient optimization, on-off fluid flow model.

23 pages (including figures)

1. Introduction

Broadband networks will support a variety of services such as voice, video, data, and image with different traffic characteristics, different qualities of service, and different bandwidth requirements and holding times in an integrated environment. Asynchronous Transfer Mode (ATM) is considered the preferred transfer mode to support various services on broadband networks. In an ATM network, traffic can be considered at the Virtual Path (VP) level, the call level, the burst level and the cell level [14]. Discussion on control issues at different levels can be found, for example, in [22], [28]. A connection-oriented transport mechanism is to be used for ATM networks. This means a virtual connection is required to be set up between the origin and the destination for a connection request. The call then uses this connection. There are two operational steps. During call setup, network resources are checked before the connection is allowed. Once the call is accepted, network traffic management monitors the traffic status and performs a policing function to ensure that resources are properly used and, accordingly, controls can be applied at different levels to satisfy quality of service.

Statistical multiplexing is possible at all levels of ATM-based networks. There are different implications due to introduction (or non-introduction) of statistical multiplexing at different levels. For example, consider a network that does not use any virtual path connection (VPC); then all the traffic can be statistically multiplexed on virtual channel connections (VCC) sharing common network links. Though this may result in minimum capacity networks, the VCC call establishment cost could be significant; further, this scenario may require complex admission control schemes. This is because each VCC would need to negotiate a connection request at each intermediate node between source and destination [10]. On the other hand, if peak rates are allocated at each level, then resources would be poorly utilized. Thus, techniques are needed to make better use of available resources while providing probabilistic guarantees of quality-of-service (QoS). Use of virtual path as an effective transport technique, and for routing and resource management for ATM networks has recently gained considerable attention [1], [2], [9], [10], [11], [13], [21], [25], [27], [32], [38], [40]. By grouping virtual circuits into a virtual path, an ATM-based network can be better managed. This can result in reduction of call set up time if there exists a virtual path (with enough capacity) between an origin and a destination since there is no need for extra processing at the intermediate nodes. This will also allow calls of similar traffic characteristics to be statistically multiplexed between an origin-destination pair and simple call admission control schemes may be needed. For ease of network management and control, virtual paths may be defined for dissimilar traffic types and can be deterministically multiplexed. This deterministic multiplexing of different virtual paths is likely to result in requirements of more network capacity than if *everything* is statistically multiplexed. However, at the same time, dynamic bandwidth control and virtual path rearrangement can be provided using VP concept with less complex admission control schemes. Thus, we consider here statistical multiplexing of similar traffic within a virtual path and deterministic multiplexing of various virtual paths along with dynamic bandwidth control and virtual path reconfiguration.

In a companion paper [34], we have outlined a framework and models for network design and management for a wide-area dynamically reconfigurable ATM network based on the virtual path concept from a network planning and management perspective. In this current paper, we present an approach for network dimensioning of such ATM networks. Our approach is based on the observation that statistical multiplexing of virtual circuits for a traffic class in a virtual path, and the deterministic multiplexing of different virtual path lead to decoupling of the network dimensioning problem into the bandwidth estimation problem *and* the combined virtual path routing and capacity design problem. Based on this observation, we present here a multi-hour, multi-traffic class network design model for providing specified quality of service in dynamically reconfigurable networks and present a decomposition algorithm by looking at the dual problem and using subgradient optimization, and provide computational results on network saving for realistic network traffic data. Multi-hour design has been used for (single service) dynamic call routing circuit-switched networks with considerable network saving [5]. The concept of dynamic reconfigurability has been addressed for circuit-switched networks [6], [7], [18], [19]. There have been some work on ATM network dimensioning ([17], [30], [31], [36], [37]) and for VP layout design [2] – all these work take different approaches than ours. To our knowledge, a multi-hour, multi-traffic class network design model for a dynamically reconfigurable ATM network based on the concept of virtual path has not been addressed to date. Our approach can be used by network designers/planners for network capacity planning of wide-area public or private ATM-based networks.

The rest of the paper is organized as follows: in the next section, we start with a discussion of the role of virtual paths in ATM network and present the multi-hour network design model. In section 3, we present a decomposition algorithm. In section 4, we present computational results for realistic network traffic data. Finally, in section 5, we summarize our approach and discuss how our approach may be useful to network designers.

2. A Network Design Model

Use of virtual path as an effective transport technique for ATM-based networks has been gaining considerable attention [1], [38]. (For discussion on the role of VP in resource management, refer to [9], [10], [22], [11], [32], [27].) A virtual path provides a logical direct link between virtual path terminators. Using virtual paths, an ATM network can be better managed by grouping virtual circuits into bundles [38]. Using VPs, flexibility in traffic management is possible due to separation of the logical transport network from the physical transmission network. For our work here, we assume that VCs of *similar* traffic characteristics and Quality of Service (QoS) requirements are statistically multiplexed on a virtual path. For brevity, we will refer to it as *STQoS types* or *classes*. Note that more than one STQoS type may be preferable between an origin-destination pair for traffic of significantly different traffic characteristics and requirements. For example, it may be minimally preferable to have an STQoS type defined for real-time services and another for non-real time services. Use of VP simplifies B-ISDN call processing and reduces call establishment time since there is no processing required at the intermediate nodes. For example, as presented in [38], the explicit scheme can be used in the cell header organization of virtual paths. This way, processing on a call-by-call basis at each intermediate node can be eliminated during call establishment and release; consequently, call set up time can be significantly reduced. Alternately, the network can be a collection of ATM switches and ATM cross-connect systems where the traffic demand between two end ATM switching nodes can be cross-connected at intermediate nodes using ATM cross-connect systems [11], [32], [34]. There are also other advantages in using virtual path concept as noted in [22]: class of service control is easier to implement than when different types of traffic are supported over the same virtual path; dynamic bandwidth control is easier to implement per path due to similar traffic characteristics; multiplexing several classes of traffic with different characteristics decreases the multiplexing gain compared to the multiplexing of traffic with the same characteristics.

Based on the above advantages of the VP concept, we assume the following for effective network design and management: we allow statistical multiplexing between calls of the same STQoS class over virtual circuits *within* the virtual path allocated for this STQoS type; however, we assume *no* statistical multiplexing between different virtual paths for different STQoS classes (i.e., only deterministic multiplexing). As mentioned before, a key advantage is that call processing time can be significantly reduced due to the introduction of VPs. In addition to the benefits that are already discussed, we note the following: each virtual path for each STQoS type is assigned a certain bandwidth which determines the number of virtual channels it can support to provide satisfactory QoS. We still use dynamic path bandwidth control to adjust bandwidth required based on anticipated demand at different times during the day. Following [9], we consider that when bandwidth is reserved on a VP, a limit is placed on the number of calls and bursts in progress on that VP at any time. This also simplifies the control decision to verify the number of calls and bursts in progress compared to allocated maximum. Further, the use of the concept of statistical multiplexing among VCs of same STQoS type in a VP and deterministic multiplexing of different VPs simplifies the non-linear optimization problem of network dimensioning to meet certain QoS guarantees for a given traffic demand and class. Specifically, it has the following major advantage: it *decouples* the network dimensioning problem into the problem of determining the bandwidth needed on each virtual path for each STQoS type *and* the associated problem of determination of virtual path routing for each STQoS type and sizing of the network. The work here is based on this key observation/advantage of *decoupling* the bandwidth estimation problem *and* the VP routing and network sizing problem.

We first discuss the second problem, i.e., our model for the *dynamic* VP routing and network sizing problem. In the framework of a network, there are a number of possible virtual paths that an STQoS type can take between two switching end points (or a demand pair). There are possible demands between any two ATM switching nodes in the network (demand pairs). Thus, we consider the determination of virtual path for each STQoS type and each demand pair in the network. Specifically, given the estimated bandwidth required for a particular STQoS type and between two ATM switching end points, we need to determine one virtual path (non-bifurcated routing) from the list of the possible (candidate) virtual paths between the end points that use ATM cross-connect nodes at intermediate points (for setting up a VP). Note that a virtual path is constructed by connecting one or more network links. Additionally, for a particular demand pair, different virtual paths may be taken by different STQoS types and further, due to network dynamism, at different times during a day (with different upper bound on bandwidth requirements to satisfy the request for connections at that time of the day). For example, consider Fig. 1. Here for the demand pair A-D in the morning, calls for STQoS type $s = 1$ may take the route A-B-D, while STQoS type $s = 2$ may take the route A-C-B-D; however, in the afternoon, the calls for both STQoS types $s = 1$ and $s = 2$ may take the VP route A-C-D. It should be understood that a VP route A-B-D means that the connection requests at ATM switch A for destination ATM switch D goes from ATM switch A to ATM cross-connect A to ATM cross-connect B to

ATM cross-connect D to ATM switch D (note that it does *not* enter ATM switch B). Due to the flow of traffic on different virtual paths at different times of the day, we can determine the total traffic on each link of the network at different times during the day by adding for different pairs and different STQoS types that use this link (this is due to deterministic multiplexing of different STQoS types). For the formulation we are about to present for the VP routing and network sizing problem, we assume that we are given a set of ATM switching and ATM cross-connect locations and a set of possible links for which capacity has to be determined (network sizing). Change of VP routes for different times during the day provides us with *dynamic* virtual path bandwidth control and dynamic VP based reconfigurability for better use of resources. Thus, to summarize, our work here considers dynamic bandwidth control with dynamic network reconfigurability based on virtual path for broadband networks such that optimal capacity on links can be provided to satisfy QoS requirements for each STQoS class and for traffic variation during the day. We assume that the day has been clustered into several load periods ($\mathcal{H}$) and we introduce a superscript $h \in \mathcal{H}$ to represent different load periods. Note that $\mathcal{H}$ need not be all the hours of a day; it can be clustered into important traffic load change periods during a day. To mathematically represent the problem, we introduce the following notation:

- $\mathcal{K}$ The set of node (demand) pairs in the network
- $\mathcal{S}$ The set of STQoS types
- $\mathcal{L}$ The set of links in the network
- $\mathcal{P}_k^{sh}$ The set of possible candidate virtual paths for STQoS type $s \in \mathcal{S}$, demand pair $k \in \mathcal{K}$ used for all $h \in \mathcal{H}$
- x_{kj}^{sh} Virtual path routing variables – 1 if STQoS type $s \in \mathcal{S}_k$, $k \in \mathcal{K}$ uses path $j \in \mathcal{P}_k^{sh}$ in $h \in \mathcal{H}$; 0 otherwise
- b_k^{sh} $= bw(A_k^{sh}, T_s, QoS_s)$, Estimated bandwidth requirement for traffic amount A_k^{sh} (with traffic descriptor T_s and quality of service requirement QoS_s) for STQoS $s \in \mathcal{S}$ for $k \in \mathcal{K}$ in $h \in \mathcal{H}$ [discussed later]
- $\delta_{kj}^{s\ell h}$ Link-path incidence matrix; 1 if path $j \in \mathcal{P}_k^{sh}$ for STQoS $s \in \mathcal{S}$ and for pair $k \in \mathcal{K}$ in $h \in \mathcal{H}$ uses link $\ell \in \mathcal{L}$; 0 otherwise
- y_ℓ Sizing (topology) variables; the number of units of high capacity on traffic link $\ell \in \mathcal{L}$
- α Capacity of a high capacity unit
- c_ℓ Cost of a high capacity unit on link $\ell \in \mathcal{L}$

The following model, to be referred to often as problem (P), can be used for multi-hour, multi-STQoS class network dimensioning of a dynamically reconfigurable VP routing network:

$$v_P = \min_{\{x,y\}} \sum_{\ell \in \mathcal{L}} c_\ell y_\ell \qquad (1a)$$

subject to

$$\sum_{j \in \mathcal{P}_k^{sh}} x_{kj}^{sh} = 1, \quad s \in \mathcal{S}, \quad k \in \mathcal{K}, \quad h \in \mathcal{H} \tag{1b}$$

$$\sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} b_k^{sh} \sum_{j \in \mathcal{P}_k^{sh}} \delta_{kj}^{s\ell h} x_{kj}^{sh} \leq \alpha y_\ell, \quad \ell \in \mathcal{L}, \quad h \in \mathcal{H} \tag{1c}$$

$$x_{kj}^{sh} = 0 \text{ or } 1, \quad j \in \mathcal{P}_k^{sh}, \ s \in \mathcal{S}, \ k \in \mathcal{K}, \ h \in \mathcal{H} \tag{1d}$$

$$y_\ell \geq 0 \text{ and integer}, \quad \ell \in \mathcal{L} \tag{1e}$$

In this model (1), the objective function (1a) represents the total capacity cost for network links. (1b) and (1d) are the decisions of choosing a virtual path for a STQoS type for a node pair at different times during the day; here, one out of several possible VP paths is chosen. The left hand quantity in (1c) is the total link capacity required at a particular load period for virtual paths set up using that link by different source-destination node pairs; Thus, constraints (1c) say that the link flow in each load period is going to force the determination of capacity on the link to cover for any time during the day. Different sets of candidate paths for each load period and service for each traffic pair may be explicitly provided through the notation $\mathcal{P}_k^{sh}$. This model takes possible candidate paths as an input; these candidate paths, for example, can be generated using a k-shortest path algorithm [29]. The idea of candidate paths is advantageous in restricting (or allowing) choice of certain paths due to information available from various network elements based on network growth, and restricting the maximum number of links allowed for a path (often the case in existing telecommunications networks) to limit the number of intermediate cross-connect ports used or to limit the delay for certain service types, and thereby, preprocessing the set of candidate virtual paths (hence the link-path formulation). Note that this approach is an integrated approach since the solution of the model provides network capacity as well as VP routing at different times of the day.

We now discuss the bandwidth (often referred to as the *equivalent bandwidth*) estimation problem; i.e., how the quantity b_k^{sh} can be estimated for each STQoS class for different traffic pairs at different load period during the day. It should be noted that although theoretically, each conceivable traffic type can be an STQoS class by itself, in practice this would generate too many classes and could become an administrative nightmare for network operations and management. Classifying various traffic types under a single STQoS class or putting them into different STQoS classes is itself a complex problem. For example, Suruagy Monteiro [42] addressed this problem in terms of whether to integrate or not (see also, [43])– this issue certainly deserves further investigation, and is, however, beyond the scope of this paper. For the purpose of the rest of the work here, we consider an STQoS class to have homogeneous traffic with same QoS requirements.

Suppose that for an STQoS class s, the grade-of-service parameter is given in terms of acceptable connection denial probability, q_s, and the offered traffic for pair k in load period h is given by A_k^{sh} erlangs. Recall that with decoupling we arrive at essentially providing a virtual link from origin to destination switching node for a STQoS class. Thus given offered load in erlangs, A_k^{sh}, and connection denial probability, q_s, we can use inverse Erlang-blocking formula (see [26]) $E^{-1}(A_k^{sh}, q_s)$, to estimate the maximum number of virtual circuits that need to be connected for this class, i.e., $N_k^{sh} = N(A_k^{sh}, q_s) = \lceil E^{-1}(A_k^{sh}, q_s) \rceil$ (here $\lceil x \rceil$ denotes the smallest integer bigger than or equal to x). (A similar discussion as above is also given by [37]). This, thus, puts a limit on the maximum number of connections to be allowed. Now if each connection for a particular STQoS type requires peak rate allocation, e.g., class-1 traffic in BISDN [8, p. 135], the traffic descriptor T_s will be the peak rate, $(R_{\text{peak}})_s$, per connection for this service class, and the bandwidth estimate is $b_k^{sh} = N_k^{sh}(R_{\text{peak}})_s$. Thus, in the case of class-1 traffic in BISDN, there is no statistical multiplexing of connections within an STQoS class. On the other hand, if an STQoS class can be characterized by an on-off model, then the traffic descriptor in this case can be given by

$$T_s = \{\rho_s, m_s, (R_{\text{peak}})_s\}$$

where ρ_s = utilization, m_s = mean_burst_period, $(R_{\text{peak}})_s$ =peak_rate ([20]). To allow N_k^{sh} connections (sources) to be admitted to the network (based on A_k^{sh} and q_s), we adapt here an approach, to be referred as InvAMS, based on the the fluid-flow approach given by [3], to estimate b_k^{sh} for each STQoS type $s \in \mathcal{S}$, pair $k \in \mathcal{K}$ in $h \in \mathcal{H}$ for the quality-of-service, QoS, given in terms of buffer overflow probability, p_s; see appendix for detail. [It is worth noting other approaches for bandwidth estimation: e.g., Guérin *et al* [20] have presented an approximate equivalent bandwidth estimation method based on the approach by Anick *et al* [3] for computation in real-time; Suruagy Montiero *et al* [43] provide an approach based on the finite buffer fluid-flow model by Tucker [44]; the capacity design for a *single* entity (link or path) based on a quality of service requirement specified by p-th percentile delay of m cells has been presented by Sato and Sato [39].]

It should be noted that if a more detailed traffic descriptor than the on-off model suits some emerging service, then an appropriate bandwidth estimation procedure has to be used instead of InvAMS procedure; note that this requires only a new module for bandwidth estimation for this emerging service *without* needing any change in problem (P). Further, in case different types of heterogeneous sources are decided to be multiplexed in an STQoS class, an approach based on the work by Elwalid and Mitra [12] may be pursued towards estimating bandwidth requirement for that class. However, for the rest of the discussion and clarity, we will refer only to InvAMS procedure with the 3-tuple traffic descriptor and consider statistical multiplexing of a homogeneous traffic type within an STQoS class.

3. A Decomposition Algorithm

In this section, we describe a decomposition algorithm for the problem (P) using Lagrangean relaxation with duality and subgradient optimization ([23]). This type of approach has been used successfully for solving data network routing and design problems (see, for example, [15], [16]), but have not been addressed in the context of multi-hour, multi-STQoS class design. Convergence results for subgradient optimization can be found, for example, in [35]. The essence of this decomposition algorithm is to consider the dual problem of (P) by relaxing constraints ($1c$) to arrive at simpler subproblems which can be solved easily, and then use updating of dual variables to iteratively solve simpler subproblems to drive towards the optimal solution of the original problem (P). First, we add an artificial upper bound on the variables y (its use will be seen in the dual problem; see(5)). Specifically, let the bound be $\hat{y}_\ell, \ell \in \mathcal{L}$, i.e.,

$$y_\ell \leq \hat{y}_\ell, \quad \ell \in \mathcal{L},$$

and in compact form, $y \leq \hat{y}$. Let $u = (u_\ell^h)$ be the dual multiplier associated with the constraint ($1c$). Then the Lagrangean is

$$L(x, y, u) = \sum_{\ell \in \mathcal{L}} c_\ell y_\ell + \sum_{\ell \in \mathcal{L}} \sum_{h \in \mathcal{H}} u_\ell^h \left(\sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} b_k^{sh} \sum_{j \in \mathcal{P}_k^{sh}} \delta_{kj}^{s\ell h} x_{kj}^{sh} - \alpha y_\ell \right). \tag{2}$$

Rearranging,

$$L(x, y, u) = \sum_{\ell \in \mathcal{L}} \left(c_\ell - \alpha \sum_{h \in \mathcal{H}} u_\ell^h \right) y_\ell \; + \; \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} b_k^{sh} \sum_{j \in \mathcal{P}_k^{sh}} \sum_{\ell \in \mathcal{L}} u_\ell^h \delta_{kj}^{s\ell h} x_{kj}^{sh}. \tag{2'}$$

The dual problem (D) is

$$v_D = \max_{u \geq 0} g(u) \tag{3a}$$

where

$$g(u) = \min_{x,y} L(x, y, u). \tag{3b}$$

Note that for given u, the Lagrangean is separable in x and y and reduces (2) to solving two independent subproblems, i.e.,

$$\min_{x,y} L(x,y,u) = \min_{y} L_1(y,u) + \min_{x} L_2(x,u) = g_1(u) + g_2(u) \tag{4a}$$

where

$$g_1(u) = \min_{y} L_1(y,u) = \min_{y} \left\{ \sum_{\ell \in \mathcal{L}} (c_\ell - \alpha \sum_{h \in \mathcal{H}} u_\ell^h) y_\ell \mid 0 \le y \le \hat{y} \right\}, \tag{4b}$$

and

$$\begin{aligned} g_2(u) &= \min_{x} L_2(x,u) \\ &= \min_{x} \left\{ \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} b_k^{sh} \sum_{j \in \mathcal{P}_k^{sh}} (\sum_{\ell \in \mathcal{L}} u_\ell^h \delta_{kj}^{s\ell h}) x_{kj}^{sh} \;\middle|\; \sum_{j \in \mathcal{P}_k^{sh}} x_{kj}^{sh} = 1,\ s \in \mathcal{S},\ h \in \mathcal{H},\ k \in \mathcal{K} \right\}. \end{aligned} \tag{4c}$$

For subproblem $(4b)$, observe that $L_1(y,u)$ is further separable to each link variable since it has only bounding constraints on the link variables, and thus the solution to $(4b)$ for each ℓ can be easily obtained by setting:

$$y_\ell^*(u) = \begin{cases} 0, & \text{if } c_\ell \ge \alpha \sum_{h \in \mathcal{H}} u_\ell^h \\ \hat{y}_\ell, & \text{if } c_\ell < \alpha \sum_{h \in \mathcal{H}} u_\ell^h. \end{cases} \tag{5}$$

Further, note that the other subproblem $(4c)$ is also separable for each k, h and s since there is no dependency constraint among k, h, s. Also, the solution is easily obtainable by setting the variable for the appropriate path j to 1 for which the "path cost", $\sum_{\ell \in \mathcal{L}} u_\ell^h \delta_{kj}^{s\ell h}$, is the least among all the paths for that specific k, h, s. We denote this path index by j^* so that $x_{kj^*}^{sh}(u) = 1$.

Note that the problem (D) is nonsmooth; as such, we use a subgradient approach to solve the dual problem [23]. This method iterates on the dual variable u. Thus, given u, once the solutions to the subproblems $(4b)$ and $(4c)$ are obtained, a dual subgradient, $\pi = (\pi_\ell^h)$, for $g(\cdot)$ is computed using

$$\pi_\ell^h = \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} b_k^{sh} \delta_{kj^*}^{s\ell h} x_{kj^*}^{sh}(u) - \alpha y_\ell^*(u), \quad \ell \in \mathcal{L}, h \in \mathcal{H}. \tag{6}$$

Then the dual multiplier, u, is updated using

$$u_\ell^h \leftarrow \max\ \left\{0, u_\ell^h + \lambda \pi_\ell^h\right\}, \quad \ell \in \mathcal{L}, h \in \mathcal{H} \tag{7}$$

where, the step size, λ, is given by ([23])

$$\lambda = \rho \frac{g^{\#} - g(u)}{||\pi||^2} \tag{8}$$

and where $g^{\#}$ is an upper bound on the dual objective and the relaxation parameter, ρ, is chosen such that $0 < \rho \leq 2$.

A primal feasible point for problem (P) can be easily obtained at each iteration by considering the solution $x^{sh}_{kj^*}(u) = 1$ to compute the left hand side of $(1c)$ so as to generate a feasible y, and in turn, we also can compute a primal objective function value. As the dual iteration progresses, we check for decrease in the primal objective value and store the best primal solution, and the best primal cost so far.

Note that if the integrality of link variables y is relaxed in the original problem (P), then we have a mixed integer programming (MIP) problem and it is clear that the optimal objective function value for this mixed integer problem, $v_{MIP} \leq v_P$. Coupling this fact with the result of the weak duality theorem [35, p. 203], we have the relation:

$$v_P \geq v_{MIP} \geq v_D.$$

Given this observation, the upper bound on the dual objective, $g^{\#}$, is set to be the lowest value of the objective function of the MIP problem as the iteration progresses. For notational convenience, we will denote the lowest network cost obtained using the decomposition algorithm for problem (P) by $\mathbf{C}_P$, for the MIP problem by $\mathbf{C}_{MIP}$, and the best dual cost by $\mathbf{C}_D$.

In our implementation, coded in C language, we set the maximum dual iterations to 1000; ρ was initially set to 2 and was halved whenever the dual objective value did not improve in 40 iterations.

To summarize, we have the following algorithmic steps to solve the entire multi-hour, multi-STQoS class network design problem.

Algorithmic steps

Step 1 Generate candidate set of virtual paths $\mathcal{P}^{sh}_k$ for $s \in \mathcal{S}$, $k \in \mathcal{K}, h \in \mathcal{H}$.

Step 2 For given QoS and the traffic descriptor for different STQoS classes, and traffic amount for different node pairs and for different load periods, compute b^{sh}_k using the InvAMS procedure for $s \in \mathcal{S}$, $k \in \mathcal{K}, h \in \mathcal{H}$.

Step 3 Size network by solving multi-hour, multi-STQoS class model (P) using the decomposition algorithm.

4. Computational Results

For computational work, we use STQoS classes, each with one homogenous traffic type as already discussed in section 2; for illustration of our approach, we limit to two STQoS classes where each class has different on-off traffic descriptor values and quality of service requirements. We use here three realistic example networks which we have used in other work [33]; these example networks are extracted from an actual voice network spanning the continental US. Topological information for these example networks are given in Table 1 (also see Figs. 2, 3 and 4). Note that in these networks a circle and a square together means that an ATM switching node is co-located (same city) with an ATM cross-connect node. Although Fig. 1 more accurately depicts the relation of cross-connect and switching nodes, we have put the same city cross-connect and switching node together here to avoid cluttering the picture. The link between two cross-connect nodes is the possible link placements for which capacity is to be determined by our approach; buffering is provided with the ATM switches. (For this architectural view, see also [11].) The maximum number of sources (connections) to be admitted for the voice traffic for these networks for three different times during a day (morning, early afternoon and late afternoon) i.e, $\#(\mathcal{H}) = 3$, are considered given based on the offered load and connection denial probability (see section 2). For brevity, we refer to these three load periods as ld-1, ld-2, and ld-3, respectively, and this service as s=1. (Certainly, VP routing can be updated more often than three times in a day; our choice of three to present computational results with VP dynamism.) Since at this point we do not have realistic traffic data available to us for any other traffic, we used a fictitious traffic for the second STQoS class (s=2). The number of connections (sources) to be admitted for this STQoS class for different traffic pairs in the network for various time of the day are generated using a uniform random number generator by picking a number between 0 and 20 % of the number of sources for the voice traffic (i.e, $N_k^{2h} = \lceil$ Uniform $(0,1) \times 0.2 N_k^{1h} \rceil$, where N_k^{sh} is the number of traffic sources to be admitted for service s, traffic pair k in load period h, and where $\lceil x \rceil$ returns the smallest integer higher than x). The total traffic in terms of number of sources to be admitted for different times of the day are listed in Table 2.

For the voice traffic (s = 1), following [41], we have used the following parameters for the traffic descriptor for packetized voice procedure: utilization (ρ) = 0.6487, mean burst period (m) = 352 ms, peak rate (R_{peak}) = 32 Kbps (for ADPCM coding); these values are used in computing bandwidth requirement using the InvAMS procedure for buffer overflow probability, p, of 10^{-4}. The parameters used for the second STQoS class are: $\rho = 0.2$, $m = 300$ ms, $R_{\text{peak}} = 300$ Kbps; the buffer overflow probability used is 10^{-7}. The traffic descriptor parameters and the QoS requirements are summarized in Table 3. Note the difference between these two traffic types: the second traffic has a peak rate which is an order of magnitude more than the voice traffic, has lower utilization and mean burst period, while the overflow probability is much more stringent for the second traffic type. We assume the buffer size (r) to be 1 Mbits for each class (see InvAMS procedure in Appendix).

Network	No. of ATM Switches	No. of Traffic Pairs, $\#(\mathcal{K})$	No. of ATM Cross--Connect Nodes	No. of Links, $\#(\mathcal{L})$
EN-1	7	21	10	14
EN-2	10	45	18	27
EN-3	15	105	23	33

Table 1: Network Topology information for example networks

Network	ld-1 s=1/s=2	ld-2 s=1/s=2	ld-3 s=1/s=2	Max. Busy Hour s=1/s=2
EN-1	1652/168	1091/139	1530/189	1661/247
EN-2	2687/263	2830/321	3226/349	3531/473
EN-3	4687/488	4082/395	4956/491	5442/691

Table 2: Summary of network traffic (in terms of total number of sources/connections to be connected) for example networks

Services	ρ	m	R_{peak}	Buffer overflow probability, p
s=1	0.6487	352 ms	32 Kbps	10^{-4}
s=2	0.2	300 ms	300 Kbps	10^{-7}

Table 3: Traffic Descriptor parameters and QoSs for two STQoS classes

Given the maximum number of connections to be admitted (and the parameters listed in Table 3), InvAMS procedure is used to compute the bandwidth requirements, i.e.,

$$b_k^{sh} = InvAMS(N_k^{sh}, \rho_s, m_s, (R_{\text{peak}})_s, r, p_s), \quad s \in \mathcal{S}, h \in \mathcal{H}, k \in \mathcal{K}.$$

We have used the modular grouping value, α, for capacity unit of a link to be 1.5 Mbps $\approx$ T1 rate. (It should be noted that this link speed is used only for illustration purpose and that our model is not limited to using this value.) Given the trend that the fiber cost is comparatively low compared to port cost at the nodes, we have computed the unit cost of links of 1.5 Mbps using 100 as the cost of each termination port and 0.1 to be the distance cost per mile. Thus, $c_\ell = 2 \times 100 + 0.1 \times D_\ell$, where D_ℓ is the distance in miles for 1.5 Mbps pipe for link ℓ. The candidate paths are generated based on distance using a k-shortest path algorithm [29]. For each k, s, h, we generated a maximum of six candidate paths for EN-1, ten for EN-2 and fifteen for EN-3.

Our computational work is aimed at addressing the following items: 1) duality gap obtained using the decomposition algorithm, 2) a measure of effectiveness of our approach compared to a local shortest path based heuristic, 3) cost saving compared to (static) virtual path based maximum busy hour network design with "worst" case and observed case. We discuss these items in some detail below.

The dual-based approach for the multi-hour (MuH) design provides us with a (best) dual objective value, $\mathbf{C}_D$, which provides us with a lower bound on the objective value for the problem (P), $\mathbf{C}_P$; we can, in turn, obtain a bound on the duality gap (in %) by using $100(\mathbf{C}_P - \mathbf{C}_D)/\mathbf{C}_D$; further, the mixed integer problem provides us with an objective function value, $\mathbf{C}_{MIP}$, that is lower than the one for problem (P) and a tighter bound on duality gap if the non-integral link capacities were acceptable.

The local shortest path heuristic (LSPH) is a fairly näive rule where the bandwidth required for a traffic pair is always routed on the shortest distance route, and then the network cost is computed based on the link flow produced due to this rule. We will denote the network cost obtained using LSPH by $\mathbf{C}_{LSPH}$. This cost is primarily used to obtain some quantitative measure on gain or effectiveness due to the decomposition algorithm compared to LSPH.

Maximum busy hour refers to the maximum load for all hours (for each k and s), i.e., $N_k^{s,\max} = \max_{h \in \mathcal{H}} \{N_k^{sh}\}$; this is usually used for *static* virtual path design so as to provide for the grade-of-service at any time during the day. We will refer to this as the maximum busy hour (MaxBH) scenario. Total network traffic for each service with maximum busy hour scenario is listed in Table 2. When the subscript h is omitted from model (1), it reduces to the maximum busy hour model (with $\#(\mathcal{H}) = 1$ and bandwidth computed for $N_k^{s,\max}$) and we can use the algorithm described in the previous section for computing the best primal and dual objective values. The cost for the MaxBH scenario when compared with the cost for MuH design obtained using the decomposition algorithm provides us with a yardstick on saving with dynamic VP routing; we refer to this saving as the observed case scenario — note that this would be the maximum saving possible if the cost obtained were the *actual* optimal values. Finally, if we compare the primal cost for the multi-hour design with the dual cost for the single-hour design we, in essence, get a feel for the "worst" case saving achieved by MuH approach since the (actual) optimal dual objective for MaxBH case provides us with the maximum possible value that the primal cost could possibly go down to if static VP routing were used.

The cost results for the example networks with multi-hour traffic and maximum busy hour traffic are tabulated in Table 4 using the decomposition algorithm of the previous section. In this table, we show the best cost for problem (P), the best cost for Mixed integer program (MIP), the best dual cost, the LSPH cost as well as the duality gap and saving of dual-based decomposition algorithm compared to LSPH. The decomposition algorithm appears to provide reasonable results as the duality gap is bounded by not more than 13 % for the objective function value for problem (P), and by not more than 5 % for the objective function value for the MIP. Furthermore, observe that our decomposition algorithm method provides between 6 % and 20 % cost saving compared to the local shortest path heuristic. In Table 5, we show the iteration number when the lowest primal value, $\mathbf{C}_P$, reported in Table 4 is obtained along with the computing time in seconds (step 2 and step 3) for running to 1000 dual iterations; this is on a DEC Alpha AXP running OSF/1 operating system (DEC Model 3000/400, 64MB main memory, *SPECfp 92* benchmark = 112.5). The computing time for generating candidate paths for all traffic pairs using the k-shortest path method (step 1) was 0.12, 0.89 and 2.70 seconds for EN-1, EN-2 and EN-3, respectively. Thus, we observe that a good solution to problem (P) can be obtained in a reasonable computing time.

	$\mathbf{C}_P$	$\mathbf{C}_{MIP}$	$\mathbf{C}_D$	$\mathbf{C}_{LSPH}$	[1] (%)	[2] (%)	[3] (%)
EN-1 (MaxBH)	15278.90	13821.93	13821.60	17568.00	10.54	0.00	14.98
EN-1 (MuH)	13402.00	12571.76	12257.29	16016.30	9.34	2.57	19.51
EN-2 (MaxBH)	57129.90	54100.36	54045.99	61875.70	5.71	0.10	8.31
EN-2 (MuH)	51485.10	47980.03	45760.05	54710.70	12.51	4.85	6.27
EN-3 (MaxBH)	91098.90	87818.22	87799.59	97790.00	3.76	0.02	7.34
EN-3 (MuH)	81599.10	77892.32	74766.10	89328.60	9.14	4.18	9.47

([1] - duality gap = $\frac{(\mathbf{C}_P - \mathbf{C}_D)100}{\mathbf{C}_D}\%$; [2] - duality gap with MIP = $\frac{(\mathbf{C}_{MIP} - \mathbf{C}_D)100}{\mathbf{C}_D}\%$; [3] - saving compared to LSPH = $\frac{(\mathbf{C}_{LSPH} - \mathbf{C}_P)100}{\mathbf{C}_P}\%$; MaxBH = Maximum busy hour; MuH = Multi-Hour)

Table 4: Cost of network design, duality gap and comparison

	Iteration # at lowest cost	Time (sec)
EN-1 (MaxBH)	1	1.43
EN-1 (MuH)	136	4.18
EN-2 (MaxBH)	136	9.13
EN-2 (MuH)	508	26.98
EN-3 (MaxBH)	114	42.33
EN-3 (MuH)	680	127.25

Table 5: Iteration number at lowest cost and computational time for example networks

	"worst" case saving (%)	observed saving (%)
EN-1	3.13	14.00
EN-2	4.97	10.96
EN-3	7.60	11.64

Table 6: Cost saving due to multi-hour dynamic VP routing design compared to Maximum busy hour static VP routing design

Finally, in Table 6, we report gain due to multi-hour design. Specifically, we found that, for the test problems, although the multi-hour design approach with dynamic bandwidth and virtual path reconfigurability may save a minimum of 3 % in the "worst" case compared to maximum busy-hour design static VP routing design, the observed case saving is between 10 and 14 % – a significant gain.

5. Summary and Discussion

In this paper, we present an approach for dimensioning wide-area ATM based networks. We consider here the scenario where traffic types of similar traffic characteristics and QoS requirements are grouped under an STQoS class for statistical multiplexing of connections using the virtual path concept while different virtual paths are deterministically multiplexed on a network link. We present a discussion to show how this concept leads to the key observation that the network dimensioning problem can be decoupled into two manageable subproblems: the bandwidth estimation problem *and* the combined virtual path routing and capacity design problem. We further consider the traffic pattern behavior due to the traffic load variation depending on the time of the day for different traffic pairs and traffic classes in the network to address a dynamically reconfigurable virtual path based ATM network environment; this is captured in the dimensioning process as well. We then present computational procedure to efficiently determine network sizing while providing computational bound on the solution quality using duality theory. We show that, for the test problems considered, our approach does between 6% to 20% better than a local shortest path heuristic. Furthermore, for the same test problems we have shown that considering network dynamism through variation of traffic during the course of a day can save between 10 to 14% in network design costs compared to a static network based on maximum busy hour traffic.

Two important observations should be noted from this work: 1) although we have used an on-off model for demonstration of the bandwidth estimation procedure, other more accurate model(s) may be used for the bandwidth estimation part depending on the type of service; importantly, this does not change the network design paradigm present here since the estimation problem is decoupled from the routing/capacity design problem, 2) our approach allows for the scenario where different node pairs may have different number of STQoS classes; this means a newly emerging service may be deployed for selected traffic pairs based on the community of interest and demand forecast since the virtual path concept is used. It is hoped that our approach will be useful to network designers involved in planning, design and deployment of wide-area ATM-based networks. The interested reader is referred to [34] for models for network servicing and monitoring of ATM networks.

Appendix

We present the InvAMS procedure here. For simplicity, consider one STQoS type and one demand pair. InvAMS procedure requires a subroutine to compute overflow probability based on the two-state, fluid flow model due to Anick *et al* [3]. Here, each traffic source is either in an idle state (no transmission) or a burst state (transmission at peak rate). We present here an approach that can be used for off-line computation as is the case with the network design phase of network planning and management. We assume that the burst and idle period are i.i.d. and exponentially distributed, then a traffic source can be characterized by the following three parameters:

ρ := utilization, fraction of time the source is active

m := mean burst period

R_{peak} := Peak rate

If $1/\lambda$ and $1/\mu$ are mean idle and burst period, then

$$\mu = \frac{1}{m}, \qquad \lambda = \frac{\rho}{m(1-\rho)}.$$

Consider N sources. If $F_i(r)$ is the equilibrium probability that i sources are active and the buffer length does not exceed r and $F_i = 0$ for $i \notin [0, N]$, then the set of differential equations governing the equilibrium buffer distribution can be given by [3]

$$(iR_{\text{peak}} - b)\frac{dF_i(r)}{dr} = (N-i+1)\lambda F_{i-1} - \{(N-i)\lambda + i\mu\}F_i + (i+1)\mu F_{i+1}, \qquad i \in [0, N]$$

where $b :=$ capacity (bandwidth). In matrix notation:

$$\mathbf{D}\frac{d}{dx}\mathbf{F}(r) = \mathbf{M}\mathbf{F}(r), \qquad r \geq 0.$$

The following eigen problem can be solved analytically [3]

$$z\mathbf{D}\phi = \mathbf{M}\phi$$

(z is some eigen value of $\mathbf{D}^{-1}\mathbf{M}$, and ϕ is the associated right eigenvector). The probability of overflow beyond r is given by

$$p = 1 - \mathbf{1}^T\mathbf{F}(r) = -\sum_{i=0}^{N-\lfloor b/R_{\text{peak}}\rfloor-1} e^{z_i r} a_i(\mathbf{1}^T\phi_i).$$

(Here $\mathbf{1}^T$ is a vector of 1's.) Thus, given N, ρ, m, R_{peak}, b, and buffer size, r, the overflow probability can be computed based on the above approach by [3]. For brevity, we denote this procedure as $AMS(N, \rho, m, R_{\text{peak}}, b, r)$.

To compute the bandwidth required for given N, ρ, m, R_{peak}, r and acceptable overflow probability $\hat{p}$ (QoS), a simple bisection scheme as given below can be used (this is what we refer to as InvAMS procedure):

```
procedure InvAMS(N, ρ, m, R_peak, r, p̂):
    Estimate b_l and b_h such that
      p = AMS(N, ρ, m, R_peak, b_h, r) < p̂ < AMS(N, ρ, m, R_peak, b_l, r)
    while ( | log(p) − log(p̂) | > δ ) do /* for some tolerance δ > 0 */
      b = (b_l + b_h)/2
      p = AMS(N, ρ, m, R_peak, b, r)
      if ( p > p̂ ) then
        b_l = b
      else
        b_h = b
      endif
    endwhile
    return(b)
```

Acknowledgement

The traffic data used in this work is based on data provided for another work [33] by Sprint Corporation and is greatly appreciated. An earlier version of this paper has been presented at the Infocom'95.

References

[1] R. G. Addie, J. L. Burgin and S. L. Sutherland, "B-ISDN Protocol Architecture," *Proceedings of IEEE GLOBECOM'88*, pp. 22.6.1-5, 1988.

[2] S. Ahn, R. P. Tsang, S.-R. Tong and D. H. C. Du, "Virtual Path Layout Design in ATM Networks," *Proceedings of IEEE INFOCOM'94*, pp. 192-200, Toronto, Canada, June 1994.

[3] D. Anick, D. Mitra and M. M. Sondhi, "Stochastic Theory of a Data-Handling System with Multiple Sources," *Bell Sys. Tech. Journal*, vol. 61, no. 8, pp. 1871-1894, 1982.

[4] A. Arvidsson, "Management of Reconfigurable Virtual Path Networks," *14th International Teletraffic Congress*, Antibes, France, pp. 931-940, June 1994.

[5] G. R. Ash, R. H. Cardwell and R. P. Murray, "Design and Optimization of Networks with Dynamic routing," *Bell Sys. Tech. Journal*, vol. 60, no. 8, pp. 1787-1820, 1981.

[6] G. R. Ash, K. K. Chan, J-F. Labourdette, "Analysis and Design of Fully Shared Networks," *14th International Teletraffic Congress*, pp. 1311-1320, Antibes, France, June 1994.

[7] G. R. Ash and S. D. Schwartz, "Network Routing Evolution," in A. Kershenbaum *et al* (eds.) *Network Management and Control*, Plenum Press, New York, pp. 357-367, 1990.

[8] D. Bertsekas and R. Gallager, *Data Networks - Second Edition*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[9] J. Burgin, "Broadband ISDN Resource Management," *Computer Networks & ISDN Systems*, vol. 20, pp. 323–331, 1990.

[10] J. Burgin and D. Dorman, "Broadband ISDN Resource Management: The Role of Virtual Paths," *IEEE Comm. Magazine*, vol. 29, no. 9, pp. 44-48, Sept 1991.

[11] M. De Prycker, "ATM Switching on Demand," *IEEE Network*, vol. 6, no. 2, pp. 25-28, March 1992.

[12] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks," *IEEE/ACM Trans. on Networking*, vol. 1, pp. 329-343, 1993.

[13] S. P. Evans, "Optimal Bandwidth Management and Capacity Provision in a Broadband Network using Virtual Paths," *Performance Evaluation*, vol. 13, pp. 27-43, 1991.

[14] J. Filipiak, "M-Architecture: A Structural Model of Traffic Management and Control in Broadband ISDN," *IEEE Comm. Magazine*, vol. 27, no. 5, pp. 25-31, 1989.

[15] B. Gavish and I. Newman, "Capacity and Flow Assignment in Large Computer Networks," *Proceedings of IEEE INFOCOM'86*, pp. 275-284, 1986.

[16] B. Gavish and I. Newman, "Routing in a Network with Unreliable Components," *IEEE Trans. on Comm.*, vol. 40, pp. 1249-1258, 1992.

[17] M. Gerla, J. A. Suruagy Monteiro and R. Pazos, "Topology Design and Bandwidth Allocation in ATM Nets," *IEEE J on Selected Area in Comm.*, vol. 7, pp. 1253-1262, 1989.

[18] G. Gopal, C. Kim and A. Weinrib, "Dynamic Network Configuration Management," *Proceedings of IEEE International Conference on Communications (ICC'90)*, pp. 295-301, 1990.

[19] G. Gopal, C. Kim and A. Weinrib, "Algorithms for Reconfigurable Networks," *13th International Teletraffic Congress*, Copenhagen, Denmark, pp. 341-347, June 1991.

[20] R. Guérin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks," *IEEE J. on Selected Areas in Comm.*, vol. 9, pp. 968-981, 1991.

[21] S. Gupta, K. Ross and M. El Zarki, "Routing in Virtual Path Based ATM Networks," *Proceedings of IEEE GLOBECOM'92*, pp. 571-575, December 1992.

[22] I. W. Habib and T. N. Saadawi, "Controlling Flow and Avoiding Congestion in Broadband Networks," *IEEE Comm. Magazine*, vol. 29, no. 10, pp. 46-53, Oct 1991.

[23] M. Held, P. Wolfe and H. Crowder, "Validation of Subgradient Optimization," *Mathematical Programming*, vol. 6, pp. 62-88, 1974.

[24] J. Hui, M. Gursoy, N. Moayeri and R. Yates, "A Layered Broadband Switching Architecture with Physical or Virtual Path Configurations," *IEEE J. Sel. Areas in Comm.*, vol. 9, pp. 1416-1425, 1991.

[25] R.-H. Hwang, J. F. Kurose and D. Towsley, "MDP Routing in ATM Networks Using Virtual Path Concept," *Proceedings of IEEE INFOCOM'94*, pp. 1509-1517, June 1994.

[26] D. L. Jagerman, "Methods in Traffic Calculation," *AT&T Bell Labs. Tech. Journal*, vol. 63, no. 7, pp. 1283-1310, 1984.

[27] R. Kawamura, K.-i. Sato and I. Tokizawa, "Self-Healing ATM Networks Based on Virtual Path Concept," *IEEE J. Sel Areas in Comm.*, vol. 12, no. 1, pp. 120-127, 1994.

[28] K. Kawashima and H. Saito, "Teletraffic Issues in ATM Networks," *Computer Networks & ISDN Systems*, vol. 20, pp. 369-375, 1990.

[29] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[30] M.-J. Lee and J. R. Yee, "A Design Algorithm for Reconfigurable ATM Networks," *Proceedings of IEEE INFOCOM'93*, pp. 144-151, March-April 1993.

[31] K. Lindberger, "Dimensioning and Design Methods for Integrated ATM Networks," *14th International Teletraffic Congress*, Antibes, France, pp. 897-906, June 1994.

[32] M. Logothetis and S. Shioda, "Centralized Virtual Path Bandwidth Allocation Scheme for ATM Network," *IEICE Trans. Comm.*, vol. E75-B, no. 10, pp. 1071-1080, 1992.

[33] D. Medhi, "A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis," *IEEE Trans. on Communications*, vol. 42, pp. 534-548, 1994.

[34] D. Medhi, "Models for Network Design, Servicing and Monitoring of ATM Networks based on the Virtual Path Concept", *Computer Networks and ISDN Systems* (conditionally accepted).

[35] M. Minoux, *Mathematical Programming – Theory and Algorithms*, John Wiley and Sons, Chichester, 1986.

[36] C. M. D. Pazos, J. A. Suruagy Monteiro and M. Gerla, "Topological Design of Multiservice ATM Networks," *Proceedings of SBT/IEEE ITS'94*, Rio de Janeiro, Brazil, pp. 385-389, 1994.

[37] H. Saito, *Teletraffic Technologies in ATM Networks*, Artech House, Boston, Mass., 1994.

[38] K.-I. Sato, S. Ohta and I. Tokizawa, "Broad-Band ATM Network Architecture Based on Virtual Paths," *IEEE Trans. on Communications*, vol. 38, pp. 1212-1222, 1990.

[39] Y. Sato and K.-I. Sato, "Virtual Path and Link Capacity Design for ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 104-111, 1991.

[40] R. Siebenhaar, "Optimized ATM Virtual Path Bandwidth Management under Fairness Constraints," *Proceedings of IEEE GLOBECOM'94*, pp. 924-928, December 1994.

[41] K. Sriram and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data," *IEEE J on Selected Areas in Comm.*, vol. SAC-4, no. 6, pp. 833-846, 1986.

[42] J. A. Suruagy Monteiro, *Bandwidth Allocation in Broadband Integrated Services Digital Networks*, Ph.D. dissertation, Report no. CSD-900018, Computer Science Department, University of California-Los Angeles, CA, July 1990.

[43] J. A. Suruagy Monteiro, M. Gerla and L. Fratta, "Statistical Multiplexing in ATM Networks," *Performance Evaluation*, vol. 12, pp. 157-167, 1991.

[44] R. C. F. Tucker, "Accurate Method for Analysis of a Packet-Speech Multiplexer with Limited Delay," *IEEE Trans. on Comm.*, vol. 36, pp. 479-483, 1988.

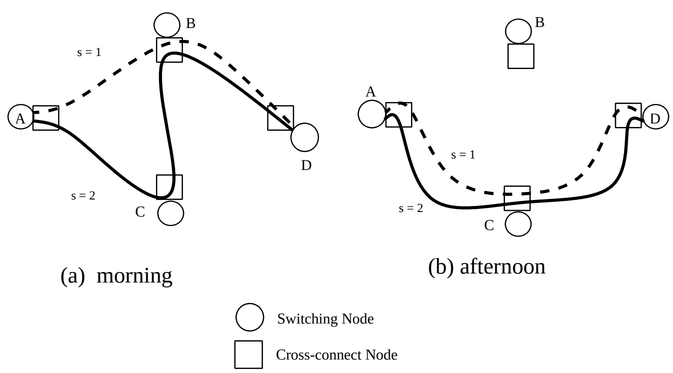

Figure 1: Virtual paths shown for two different STQoS classes for a demand pair at two different load periods during the day

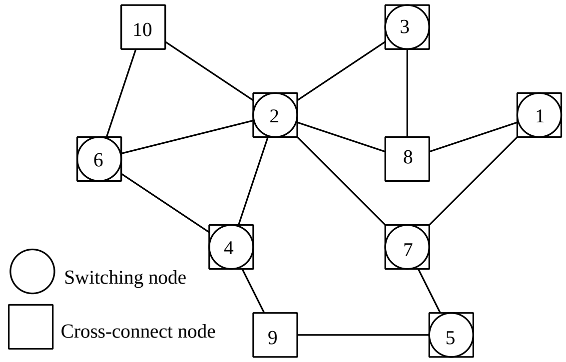

Figure 2: Topology of EN-1 (7 ATM switching nodes, 10 ATM Cross-connect nodes)

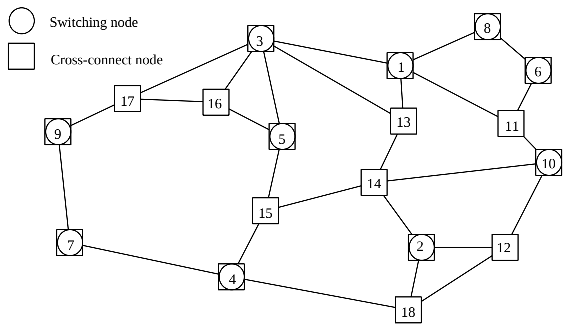

Figure 3: Topology of EN-2 (10 ATM switching nodes, 18 ATM Cross-connect nodes)

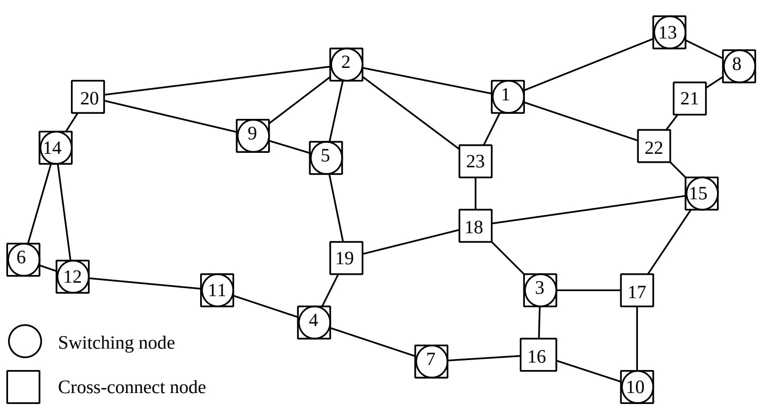

Figure 4: Topology of EN-3 (15 ATM switching nodes, 23 ATM Cross-connect nodes)