Integrated Physical and Logical Layer Design of Multimedia ATM Networks

Soumyo D. MOITRA¹, Nonmember, Eiji OKI¹¹, and Naoaki YAMANAKA††, Members

SUMMARY This letter proposes an integrated approach to multimedia ATM network design. An optimization model that combines the physical layer design with the logical layer design is developed. A key feature of the model is that the objective to be maximized is a profit function. It includes more comprehensive cost functions for the physical and logical layers. A simple heuristic algorithm to solve the model is presented. It should be useful in practice for network designers and operators. Some numerical examples are given to illustrate the application of the model and the algorithm.

Key words: ATM, network design, optimization

1. Introduction

Now that ATM is being increasingly deployed, it is important to consider integrated models for developing and provisioning ATM networks so that they will support new, multimedia services efficiently. By integrated models we mean models that try to optimally configure both the physical layer and the logical layer of ATM [1]. A major advantage of ATM is that it can configure the logical layer very efficiently, and methods to do this has been widely reported, for example in [2], [5], and [6]. Most of these methods assume sufficient capacity at the physical layer level, but in realistic situations, that might not be always feasible. With an integrated provisioning model, the physical layer will be optimized taking the ATM configurability into account, and the logical layer configurations should be more efficient.

The approach described here brings together several lines of research in network and traffic planning. We build on work done on ATM network planning strategy [1], [2], [18], on optimally designing the physical layer [8]–[10], [14], [15] and on optimally configuring the logical layer of ATM networks [5], [6], [12], [13]. The aim is to see how the advantages of ATM can be used to save on investments in the network during its life cycle while providing the required quality of service [12].

When planning networks, we only have forecasts of future demand, but actual demand may be different. In such situations, stochastic programming methods may be adopted to plan the physical capacity expansion. When actual demands are higher than planned capacity, we propose a 2-stage model that is analogous to the stochastic programming with recourse approach [11], [20].

Another related issue is setting up the objective function for optimal network planning. Traditionally a simplified cost function was used as the objective function to be minimized [8]–[10], [14], [15]. An alternative network cost function for ATM has also been proposed that is based on VP link and node costs [22]. There may also be cases when revenues need to be maximized [16], [17]. However, in the current context of telecommunications, it is probably more important to maximize profits, [2], [3]. That is, minimizing cost or maximizing revenues is not always sufficient, and network planners may prefer maximizing profits in optimizing their networks.

Among the issues that need further study are: i) developing an integrated optimization model for both physical and logical layer planning; ii) using a profit function as the objective to be maximized; iii) utilizing the concept of programming with recourse (2-stage optimization) when faced with uncertainty; iv) extending logical layer configuration methods to include capacity constraints of the physical layer and different demands among end points.

This letter suggests approaches that might help in the above areas. The objectives are to outline methods for efficiently planning multimedia (multi-QoS) ATM networks by developing a model that links the physical layer provisioning with the logical layer configuration. This should lower the cost of the physical network while making the planning process simpler. It should also help in more efficient logical layer configurations since more general cases may be considered with this and the configuration algorithm is simpler than conventional ones. Furthermore, by introducing a general profit function as the objective, we reflect the priorities of many network operators more appropriately.

By proposing a 2-stage planning process for the physical layer, we extend previous network capacity expansion models to handle unexpectedly higher demands more cost-effectively.

The solution methods proposed here for this more general model are also important because they provide simple and quick algorithms that can easily be used
by network designers and operators to quickly build multimedia-ATM networks.

We consider a two-stage network development model. The first stage provisions the network for meeting the expected demand in the first period, and the second stage does the same for demand expected in the next period. We assume a set of locations or node positions and a set of demands for multimedia services between the locations. Then we set up an optimization model to design the physical network of nodes and links to meet this demand. This is the first stage and the first period of network operation commences upon its completion.

As the demand for multiple services increases, the logical layers (VPs and VCs) are reconfigured to optimally meet these demands. This can be done by any one of the routing algorithms available. If the demands during and up to the end of the first period do not exceed the projected values, the routing of the traffic will be optimal throughout the period and the physical capacity will not be a binding constraint. Even if actual demand exceeds projections by a small amount, feasible routes may still be found, although they will no longer be optimal in comparison with a network with infinite-capacity links. However, as demand keeps increasing over time, beyond a certain point the physical network capacity in any real situation will be exhausted, and further call requests will have to be refused until capacity is expanded, that is, after the second stage.

In the second stage, the capacity of the physical network is expanded to meet the new demands expected during the second period. Then during the second period, demands for different multimedia services may again increase, and the logical layers are again reconfigured optimally, as before, given the greater physical capacity. Beyond the second period, the changes in technology will probably necessitate a completely new model with different constraints and parameters.

We note that we are assuming an ATM scenario where the logical reconfiguration can always be done easily and optimally. Thus this model is specific to multimedia ATM networks.

### 2. Model Development

In this section, we develop the integrated model for designing ATM networks. This is the main focus of this letter. The integrated approach should result in a more cost-effective or profitable ATM network than would be the case if the physical and the logical layers were planned separately.

#### 2.1 Notation

We shall use the following notation. Additional notation will be introduced as needed.

- **Data**
  - \( n \) number of nodes,
  - \( i, j, k, l, m \) indices for nodes, in \( \{n\} \),
  - \( s, d \) indices for source and destination nodes in \( \{n\} \),
  - \( k \) index for service or traffic class,
  - \( v \) index for a VP path,
  - \( \{v\} \) set of all possible VP paths among the \( n \) nodes,
  - \( DMD_{sdk} \) demand between nodes \( s \) and \( d \) for service of type \( k \),
  - \( DMD_{sd} \) total demand between nodes \( s \) and \( d \),
  - \( r_k \) unit revenue from carrying type \( k \) traffic,
  - \( D_{ij} \) distance between nodes \( i \) and \( j \),
  - \( a_{ij} \) availability of link \( ij \),

- **Decision Variables**
  - Physical Layer:
    - \( Y_{ij} \) = 1 if link \( ij \) exists, = 0 otherwise
    - \( Z_{sdk}^{ij} \) = 1 if \( DMD_{sdk} \) is routed along link \( ij \), = 0 otherwise
  - Logical Layer:
    - \( V_{vk}^i \) indicator variable, = 1 if the VP \( v \) is configured for service \( k \), = 0 otherwise
    - \( U_{sdk}^i \) = 1 if \( DMD_{sdk} \) is routed along \( v \), = 0 otherwise
    - \( W^i_{v} \) = 1 if \( v \) uses link \( ij \), = 0 otherwise

- **Derived variables**
  - \( X_{ijk} \) traffic of type \( k \) that is routed along link \( ij \),
  - \( X_{ij} \) total traffic that is routed along link \( ij \),
  - \( L_{ij} \) capacity (bandwidth) of link \( ij \) (= \( X_{ij} \) initially),
  - \( T_{sdk}^{ij} \) = \( Z_{sdk}^{ij} \cdot DMD_{sdk} \),
  - \( N_i \) capacity of node \( i \),
  - \( F_{sdk}^{ij} \) = \( DMD_{sdk} \cdot \sum_v U_{sdk}^i \cdot W^i_{v} \), the flow along link \( ij \) due to demand \( DMD_{sdk} \),
  - \( BW^k_v \) bandwidth allocated on \( v \) to service class \( k \),
  - \( BW_v \) = \( \sum_k BW^k_v \), capacity of \( v \),

- **Other variables**
  - \( \nu \) total number of links in the network,
  - \( TRF \) total traffic actually carried on the network,
  - \( TCP \) total cost of the physical network
  - \( TCL \) total cost of the logical network

#### 2.2 Physical Layer Design—Stage 1

As discussed in the Overview, this is the stage when the physical layer is initially planned. We shall refer...
to this stage as PL1. The decision variables are $Y_{ij}$, which is an indicator variable depending on whether link $ij$ is installed, and $Z_{bsd}^{ij}$, which depends on whether $DMD_{bsd}$ is routed on link $ij$. $N_i$ is the capacity of node $i$. We consider the following objective function:

$$\text{Maximize } \phi = \sum_k r_k \cdot \text{TRF}_k - TCP, \quad (1)$$

where $\phi$ is a profit function.

The first term on the right is the total revenue, which is the product of the traffic carried ($\text{TRF}_k$) and unit revenue ($r_k$) summed over all traffic classes $k$. The second term is the total cost of the physical layer.

While the revenue term appears simple, its evaluation requires knowledge of the charges made for carrying traffic for each traffic class and knowledge of the actual traffic carried by class. This form has been used in most cases in the literature where revenue has been considered [2], [16], [17]. The above formulation has also been used in an actual application on telecommunications network dimensioning [3]. It is important to note that $\text{TRF}_k$ is the carried traffic for which customers are billed and for which data will therefore be available.

In general,

$$\text{TRF}_k = \sum_{sd} DMD_{sdk} \cdot a_{sdk} \quad (2)$$

where $a_{sdk}$ is the availability of the network for a demand group $sdk$ [16], [17]. In this analysis, we will assume a highly reliable network with all availabilities $a_{sdk}$ equal to 1. Then, with this approximation, the objective becomes:

$$\text{Maximize } \phi = \sum_k r_k \cdot \sum_{ij} DMD_{ijk} - TCP \quad (3)$$

where $TCP$ is derived in detail below.

We also assume that the physical layer must be designed to meet the projected demand. Since unit revenues are given, the first term will be a constant and, as discussed later, can be omitted for the optimization procedure. The objective function then becomes:

$$\text{Minimize } TCP \quad (4)$$

subject to the following constraints:

$$\sum_j T_{sdk}^{ij} - \sum_j T_{sdk}^{ji} = \begin{cases} DMD_{sdk} & \text{if } i = s \\ -DMD_{sdk} & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

with $DMD_{sdk} \geq 0$, and $Z \leq Y$, $Y = 0$ or 1 for all $i, j, l, s, d, k$.

This is a straightforward shortest path/multicommodity flow formulation that has often been used in operations research and network analysis [8], [14], [18], [21], among others. The demands are estimated from an equivalent bandwidth algorithm for VBR traffic and CBR traffic demands are translated into the same uniform units. A feasible solution to the above problem can be easily found by a greedy, shortest path algorithm. Note that this will not be optimal because of the cost function we are using, which is not based simply on unit cost per link.

The total cost $TCP$ of carrying all the traffic is derived as follows.

The total cost $TCP$ is:

$$\text{total cost } = \text{fixed cost } + \text{variable costs} \quad (6)$$

1. fixed costs = link installation + node capacity provisioning costs.
2. variable costs = distance/capacity-dependent costs + recurring costs
   a. distance/capacity-dependent costs (links and nodes)
   b. recurring costs = operations + maintenance (links + nodes)

Therefore we can write,

$$TCP = c_f \cdot \sum_{ij} Y_{ij} + c_n \cdot \sum_i N_i + c_l \cdot \sum_{ij} L_{ij} + c_d \cdot \sum_{ij} D_{ij} + c_e \cdot \sum_{ij} D_{ij} \cdot L_{ij} \quad (7)$$

where the cost coefficients can be interpreted as follows:
- $c_f$ fixed cost of installing a link,
- $c_n$ unit cost of installing capacity in a switch,
- $c_l$ unit cost of installing capacity on a link,
- $c_d$ cost of installing a link per unit distance,
- $c_e$ carrying cost of unit traffic per unit distance.

The above cost model includes all the possible cost components that might be relevant. Thus we take into account the cost of laying cables, the distances over which cables are laid, the capacities of the links, and the operations and management costs which we include in carrying costs. The fixed costs of installing the nodes are not included since it will be a constant, given that nodes will be installed at all the end locations. We assume that the cost coefficients are the same regardless of the particular link. If costs vary from link to link, then the coefficients can be made functions of each link $ij$, but the data requirements would increase greatly. Whether the improvements in the solution would justify a much more cumbersome model will have to be decided for each design problem. Our approach can be applied in either case. The step increments in link capacity expansion costs due to installation of new links are included, but possible step increments in cost as the capacity of a link increases are approximated by an straight line slope. This implies that the cost of installing unit additional capacity is constant, and hence total cost increases linearly with capacity. In general this would be appropriate and this form has been used widely in the literature [8], [14], [18], [23]. However, if it is necessary in a particular case to consider step increments explicitly, we should replace $c_l \cdot \sum_{ij} L_{ij}$ by...
\[ \sum_{ij} c_i(L_{ij}), \] where the function \( c_i(L_{ij}) \) would incorporate the step increments \([9]\). Also, we assume that no nodes are installed other than at the locations originally given. This cost function is a very general one and other cost models in the literature \([4],[8],[10],[15]\) are subsets or special cases of it.

2.3 Logical Layer Configuration

After we have a solution for the physical network design, we proceed to optimally configuring the logical layer. We shall refer to this stage as LL. Here we find the optimal VP configuration that will meet the demand. This will have to be done dynamically, as demand changes \([12],[15]\). We consider the same general objective function as before:

\[ \text{Maximize } \phi = \sum_k r_k \cdot TRF_k - TCL, \quad (8) \]

but with a different cost function \( TCL \). The cost model is based on the following considerations:

- The number of VP's used: This has an impact on cost because having to configure a VP and operating it involves a significant additional cost. Once a VP is set up however, the cost of assigning more traffic on it is assumed to be marginal. If the traffic can be transported by fewer VPs, it will be cheaper. This component of the cost is given by \( cv \sum_{ij} I_{ij}^k \), where \( cv \) is the cost of configuring a VP and making it operational.

- The number of physical links used: This is again the cost difference between using a link (that is equipping it fully for operations) and not using a link. We again assume that once a link is in operation the incremental cost of adding more traffic on it is negligible. This cost component is given by \( cu \sum_{ij} I_{ij} \) where \( I_{ij} \) is an indicator variable that is 1 if link \( ij \) is used, and \( cu \) is the cost of utilizing a physical link.

- The number of nodes being used in the network: There is a step difference in costs if a node is made operational. If the traffic can be transported by fewer nodes, it will be cheaper. Once operational, the incremental cost of handling additional traffic may or may not be significant. In this model, we assume that all end nodes or locations will have to have a switch (and they represent fixed costs which are not included in the optimization function), and that intermediate nodes will not be required. (If intermediate nodes are required, this factor will be non-zero.) We consider two different types of nodes: switches (SW) and cross-connects (XC). Each will have different costs. This is given by \( \sum_t cx_t \cdot n_t \) where there are \( n_t \) nodes providing the function of type \( t \), and \( cx_t \) is the setup cost for a node of type \( t \).

- The processing costs of traffic by class at the switches: Different classes may have different cost associated with them, depending on their QoSs, so the mix of traffic at a switch will impact on the switch operations cost. This is given by \( \sum_{jk} cp^k \sum_{ij} X_{ijk} \) where \( cp^k \) is the unit cost of processing traffic of class \( k \). This is the only capacity related cost that will be significant because it depends on the traffic classes. Other capacity related costs are dependent on the total volume and that is a constant (= total demand or total capacity, whichever is less) and hence they need not be included in the objective function during optimization. Of course, these constant costs should be calculated and added to the other costs that are found from the solution. This is necessary for computing the profit. The magnitude of these constant costs will depend on the total traffic carried and the total operational costs. The computation of these constant costs would require further cost data pertaining to that particular network. In this letter, we focus on those costs that are to be minimized since the solution provided by our proposed model and algorithm would be essential in any case.

Therefore the total cost at the logical layer is:

\[ TCL = cv \sum_i V_i + cu \sum_{ij} I_{ij} + \sum_t cx_t \cdot n_t + \sum_{jk} cp^k \sum_{ij} X_{ijk} + \sum_t cs_t \cdot N_t^k \quad (9) \]

To derive the flow constraints, we consider the set of all possible VPs that can be configured within the network. This is given by \( \{v\} \), and the total number of paths \( |v| \) is \( n \times (n - 1)/2 \) since the maximum number of unique paths from each node to every other is given by a full mesh. Henceforth, by path, we will mean a VP. The index of the VPs, \( v = (1 \text{ to } |v|) \). As defined above, \( V^k = 1 \), if the path \( v \) is configured as a VP for service class \( k \). The mapping of demand groups on the VPs is done by \( U^v_{sdk} \), which is 1 if \( DMD_{sdk} \) uses path \( v \). The mapping of the VPs on the physical links is given by \( W_{ij}^v \) which is equal to 1 if path \( v \) happens to include link \( ij \). The \( W \)'s are derived from the path configurations. Thus if path \( 5 \) is a path from link 2 to 4 via 3, then \( W_{23}^{25} = 1 \), and \( W_{24}^{23} = -1 \). We can now define the flows on the links due to the demand groups as follows:

\[ F_{sdk}^{ij} = DMD_{sdk} \sum_v U^v_{sdk} \cdot W_{ij}^v, \quad (10) \]
and therefore the flow constraints for the physical layer are:

\[ \sum_j F_{s dk}^{ij} - \sum_j F_{s dk}^{ji} = \begin{cases} DMD_{sdk} & \text{if } i = s \\ -DMD_{sdk} & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \]

(11)

The capacity constraints are:

\[ \sum_{sdk} F_{sdk}^{ij} < L_{ij}, \text{for all } i, j \]

(12)

that is, the total traffic routed along link \( ij \) cannot exceed link capacity. We can also derive \( BW_k = \) the bandwidth on path \( v \) allocated to traffic of class \( k \).

At first the network is over-engineered, and has enough capacity relative to initial demand since it has been planned to meet much higher future demands. At this stage we have \( n \) nodes, and \( \{L_{ij}\} \) links from the PL1 design stage. Since all demand can be met, revenues will be constant, hence profit maximization in this phase will be equivalent to cost minimization. As demand keeps increasing, at some point, the capacity constraints will start to become binding. However, for a subsequent period, the demand can still be satisfied, but by inefficient routing (to overcome individual link constraints). Under these conditions, costs will tend to increase very rapidly as more traffic is routed. Then, as demand increases still further, new connection requests, and/or new customers will have to be refused.

In this situation, the traffic that can be carried by the network is less than the demand (or offered load). In this phase, utilizing the profit function as the objective will enable us to allocate limited network resources to the traffic so as to yield the highest profit. This could not have been achieved without the profit function as the objective.

Also, in this phase (11) will have to be modified since all of the demands may not be necessarily satisfied. In this case the flow constraints should be

\[ \sum_j F_{s dk}^{ij} - \sum_j F_{s dk}^{ji} = \begin{cases} DMD'_{sdk} & \text{if } i = s \\ -DMD'_{sdk} & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \]

(13)

where \( DMD'_{sdk} \) is that part of the demand that is carried by the network. \( DMD'_{sdk} \) is computed in the course of executing the algorithms described in Sect. 3.2.

2.4 Physical Layer Design—Stage 2

Finally at one point, as demand increases still further, and perhaps in unanticipated ways, the connection refusal rate will become unacceptably high. Then the network operator will have to increase the physical capacity of the network, as pointed out in [1]. We shall refer to this stage of expansion of the physical layer as PL2. At this stage we propose the strategy of increasing the capacities on selected links and nodes while keeping the rest of the network the same. This may be an effective and viable option to follow if the technology allows it, for example with optical networks, where it might be easier to install more powerful lasers and receivers than to lay new cables. Similarly, expansion cards may be added to the nodes if necessary. Wave division multiplexing may also be employed to achieve this. With this strategy, we propose using the “shadow prices” on the link capacity constraints to select which links should be expanded. The shadow price of a link is given by

\[ \psi_{ij} = \frac{\partial \phi}{\partial L_{ij}} \]

(14)

and it is the increase in profits that would be realized if one of the constraints could be relaxed by one unit. The higher its shadow price, the better it is to try to relax that constraint because the higher will be the resulting increase in profits. The shadow price is 0 when the constraint is not binding (there is spare capacity on it) and so there is no point in relaxing that constraint. For a more detailed description of shadow prices, the reader can consult any standard operations research text, such as [21]. It has been used in telecommunications in [7]. To implement this strategy, we can order the links according to their shadow prices, and start increasing their capacities starting with the highest values. This can be the most cost effective way to increase network capacity for a limited period of time. However, after some point, it may be found that it is optimal to install new links (and/or intermediate nodes) to meet the increasing traffic, and the shadow price computations will indicate that. At that point, the network planner may use one of the standard network capacity expansion models. These have been well described in the literature [8], [9], [14], and therefore we do not discuss them here.

3. Solution Methodology

The focus of this letter is the development of a model that integrates the design of the physical layer with the design of the logical layer and this was done in the previous section. This section describes one possible solution methodology. The purpose of this letter is not to investigate the most efficient algorithms that can be developed to solve such problems. Therefore, we develop some simple heuristic algorithms to demonstrate how ATM network design problems can be solved using the integrated approach. The advantage of this methodology is that it provides a quick and simple procedure that gives feasible solutions that are reasonably good.
3.1 Algorithms to Solve the PL1 Problem

To solve the problem PL1, we develop a heuristic in two steps. In the first step we find an initial feasible solution (IFS), and in the next step, we improve upon it to find a “near-optimal” solution. The resulting physical network will be deliberately left over-engineered, because of some practical advantages of having spare capacity in the network. However, our heuristic allows the designer to control the degree of over-engineering that will exist.

3.1.1 Initial Feasible Solution: IFS

Initialize all link capacities to 0: Add all demands over the different types, that is, compute $DMD_{ij} = \sum_k DMD_{ijk}$, for all $ij$: Set all the link capacities to the point-to-point demands (O-D pairs): $L_{ij} = DMD_{ij}$. If $L_{ij} > 0$ set $Y_{ij} = 1$.

Now we have a mesh-like network where all demands are met by simply having a link for every S-D pair for which there is a non-zero demand. In the extreme case it may be a full mesh. This is the IFS.

3.1.2 Iterative Improvements

Our heuristic improves the IFS by eliminating links. The rationale for this is that by providing a direct link to serve every S-D pair, the IFS has too many links, resulting in too high a cost. The key to making improvements is the metric $L_{ij}/D_{ij}$. This is because we would like to eliminate links that carry very little traffic (and reroute that traffic), and also those links that are very long. Long links are costly, and the chances are that an alternative route will not be that much more costly. Both these factors suggest that the lower the value of the $L_{ij}/D_{ij}$ metric, the more likely it is that we shall be able to reduce costs by eliminating that link. Therefore we sort the links in the IFS in increasing order, and start eliminating links one at a time from that list. The elimination process is stopped according to the following stopping rules. All the rules are to be applied, and any one of them can stop the iterative process.

1. After elimination a link and rerouting the traffic on the next shortest path, TCP is recalculated. If the new TCP is higher, we keep the current link, and go on to the next link.
2. At each rerouting, we check if the next shortest path has more than two links. If so, we try further alternative routes. If no alternative route with less than three links exists, we keep the current link and go on to the next link. (That is, if at any step, the number of hops for the rerouted traffic is greater than 2, keep the original link.

3. We derive three more optional stopping rules to allow quicker termination of the algorithm if desired.
   a. If the number of remaining links is equal to $n_{min}$, stop. Here $n_{min}$ can be set by the user.
   b. If the number of iterations (= number of links already eliminated) is equal to $n_{lim}$, stop. Here $n_{lim}$ can be set by the user.
   c. If the $L_{ij}/D_{ij}$ value of the next candidate is greater than some $u_{lim}$, stop. Here $u_{lim}$ can be set by the user.

The result of this step is a reasonably sparse network, that is still over-engineered. If the degree of over-engineering is to be reduced, further iterations of link elimination can be carried out. That is, the stopping rules can be changed, or the 2-hop upper limit can be relaxed.

3.2 Algorithms for the LL Configuration

The logical layer configuration problem as we have formulated here is a linear integer program, and can be solved by a good integer programming package for medium-sized networks. The model can be extended to accommodate routing parts of a demand group on different VPs, but at the cost of increasing the number of variables. Essentially this is a combinatorial problem that is known to be NP-hard. Therefore, a heuristic will be necessary for solving very large problems, or to solve medium-sized problems quickly. While a heuristic may not give an optimal solution, it may give acceptable and practically useful solutions, as demonstrated in [10], [15]. Therefore, we develop the following heuristic algorithm:

3.2.1 Initial Routing

First we employ the round-robin (R-R) allocation scheme for the initial routing. In practice, this can come very close to the optimal routing. This scheme allocates an incremental amount of bandwidth $AL$ to each VP $v$ that connects a source-destination pair of nodes between which there is a demand. $AL$ is adjusted according to the residual bandwidth on a VP such that the physical capacities of the links are not exceeded. We assume without loss of generality that $r_1 > r_2$. The procedure is as follows:

Initialization - Setting the value of $AL$:
$AL = \min[(\min_{ij} L_{ij}), (\min_{sd} DMD_{sd1})];$
for $k=1$
for all $sd$ pairs
   if $DMD_{sd1} > 0$
      allocate $\min[AL, DMD_{sd1}]$ to the $v$ that connects $sd$
      decrease $DMD_{sd1}$ by $AL$
   while: there is capacity on $v$,
repeat previous step;  
if capacity on $v$ is exceeded, reduce $AL$ to  
residual bandwidth on $v$  
track total bandwidth allocated to $v$  
if $DMD_{sd}$ is exhausted, next $sd$  
repeat for $k=2$  
create table of allocated demands and unmet  
demands.

3.2.2 Resource Directive Approach

We now attempt to improve upon the solution obtained  
above by using modified subgradients. This method  
of optimizing network flows has been used successfully  
in the past [19]. However, the conventional methods  
are not appropriate in this case because we are using  
profit as the objective to maximize and the  
conventional methods are for cost minimization objectives  
[19]. Therefore we propose a modified method based on  
the same idea. This idea is to shift demand allocation  
among the VPs so that either i] costs are reduced while  
keeping revenues constant, or ii] revenues are increased  
with cost remaining constant. The algorithm requires  
a parameter which specifies the amount of bandwidth  
that will be exchanged between two VPs at any step,  
and we denote it by $\theta$. We implement it so that a  
demand group that uses multiple links is given lower pri-  
ority than a demand group that uses only one of those  
links. This is done with the following algorithm.

Initialization  
$\theta = 1$;  
for $k = 1$  
for each $sd$ pair  
if residual $DMD_{sd}$ > 0  
check if there is a single link $v$ that connects $sd$  
if: $v$ is that single link VP;  
then: search for all other VPs sharing $L_{sd}$  
with $BW^k > 0$  
if such a subset exists, find $v'$ which uses  
the maximum number of links in it;  
let $(lm)$ be the terminating nodes of $v'$  
subtract $\theta$ from the $DMD_{lmk}$; and  
add $\theta$ of $DMD_{sdk}$ to $v$;  
for every other link $(ij)$ used by $v'$,  
add $\theta$ to any residual $DMD_{ijk}$;  
repeat last three steps while  
$\sum_v BW^v_{ij} * W^v_{ij} < L_{ij}$.  
update table of allocated demands  
repeat for $k = 2$

The above heuristic algorithm is much simpler than  
most existing ones in the published literature, but based  
on the numerical examples worked out in the next section, it appears that the algorithm would probably give  
reasonably good solutions in practice. Thus it would  
make the task of designing the ATM logical layer much  
easier for network operators.

This completes our description of the solution algo-  
rithms. We have described how the physical layer can  
be designed, and how the logical layer can be configured  
to route demands. The expansion of the physical layer,  
may be based on the method of shadow prices. However,  
the algorithm for that stage is beyond the scope of  
this letter, and is a matter for future work. We now  
illustrate the above method with some examples.

4. Numerical Examples

4.1 Example 1

In this example, we apply the above algorithms to a  
4 node network, with 2 traffic classes or service types.  
We generate the network data randomly. The initial  
data that is required are the internode distances and  
the forecasted demands ($DMD1$ and $DMD2$ for the two  
classes). For this example, we have generated the fol-  
lowing data:

<table>
<thead>
<tr>
<th>$DMD1$</th>
<th>$i$</th>
<th>$j$</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$DMD2$</th>
<th>$i$</th>
<th>$j$</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>14</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distances</th>
<th>$i$</th>
<th>$j$</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>35</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>20</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The sample set of values for the cost coefficients used  
in computing the objective function (TCP) are:

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_f$</td>
<td>2000</td>
</tr>
<tr>
<td>$c_n$</td>
<td>200</td>
</tr>
<tr>
<td>$c_l$</td>
<td>100</td>
</tr>
<tr>
<td>$c_d$</td>
<td>800</td>
</tr>
<tr>
<td>$c_c$</td>
<td>10</td>
</tr>
</tbody>
</table>

Based on this input data, we get the IFS, with link  
capacities $L(ij)$ as:

<table>
<thead>
<tr>
<th>$L_0$ matrix</th>
<th>$i$</th>
<th>$j$</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>14</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
TCP for this network is 132,110.
From the algorithm, we eliminate link (12), and reroute its traffic on (1-4-2). Then we stop because of
the connectivity requirement. The final PL is given by:

\[
L_1 \text{ matrix }
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 0 & 0 & 20 \\
  2 & 14 & 33 \\
  3 & 0 \\
\end{array}
\]
TCP for this network is now 121,110. Thus the algorithm has reduced the total cost.
Now we have built the physical network which cannot be changed. Given any unexpected, actual demand, we will have to do the best we can given this physical network. This will be done by the LL algorithm.
The unexpected demands (NewDMD1 and NewDMD2), generated randomly so that their means are greater than the original forecasts, are:

**NewDMD1**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 3 & 2 & 3 \\
  2 & 1 & 0 \\
  3 & 2 \\
\end{array}
\]

**NewDMD2**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 9 & 0 & 9 \\
  2 & 14 & 25 \\
  3 & 0 \\
\end{array}
\]
Now we apply the LL algorithm:
a) In the R-R allocation method, we have:
i) \( k = 1 \)

\[
\begin{array}{ccc}
  i & j & v & BW^v_k \\
  12 & 1 & 3 \\
  13 & 2 & 2 \\
  14 & 3 & 3 \\
  23 & 4 & 1 \\
  24 & 5 & 0 \\
  34 & 6 & 2 \\
\end{array}
\]

ii) \( k = 2 \)

\[
\begin{array}{ccc}
  i & j & v & BW^v_k \\
  12 & 1 & 6 \\
  13 & 2 & 0 \\
  14 & 3 & 6 \\
  23 & 4 & 9 \\
  24 & 5 & 20 \\
  34 & 6 & 0 \\
\end{array}
\]
This results in all residual (spare) capacity going to 0, and residual demand as:

**Residual NewDMD2**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 3 & 0 & 3 \\
  2 & 5 & 0 \\
  3 & 0 \\
\end{array}
\]
That is, all of NewDMD1 has been met, but some NewDMD2 (lower priority traffic) is left unsatisfied.
b) Applying the modified sub-gradient algorithm:
(Take away 3 units from VP 1 \((v = 1)\) and route 3 units of residual NewDMD2 on \(v = 3\) and also on \(v = 5\).)
Final residual demand is:

**Residual NewDMD2**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 6 & 0 & 0 \\
  2 & 5 & 0 \\
  3 & 0 \\
\end{array}
\]
No more candidates for the sub-gradient method are present, so this is the best solution with the algorithm.

4.2 Example 2
For this example we also randomly generate a 4-node network to serve two traffic classes:

**DMD1**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 0 & 6 & 15 \\
  2 & 0 & 7 \\
  3 & 12 \\
\end{array}
\]

**DMD2**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 31 & 19 & 40 \\
  2 & 39 & 0 \\
  3 & 20 \\
\end{array}
\]

**Distances**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 42 & 21 & 26 \\
  2 & 25 & 12 \\
  3 & 14 \\
\end{array}
\]
From the algorithm, we get the final PL (after 2 iterations) as:

**L_2 matrix**

\[
\begin{array}{ccc}
  i & j = 2 & 3 & 4 \\
  1 & 0 & 56 & 55 \\
  2 & 77 & 0 \\
  3 & 39 \\
\end{array}
\]
TCP is reduced from 266,140 for the IFS to 241,070 for the final network.
As before, we have now built the physical network which cannot be changed. Now the unexpected demands (NewDMD1 and NewDMD2), generated randomly so that their means are again greater than the original forecasts, are:
Applying the LL algorithm:

a) In the R-R allocation method, we have:

\[
\begin{array}{c|cccc}
  i & j = 2 & 3 & 4 & \\
  1 & 0 & 0 & 6 & \\
  2 & 11 & 7 & & \\
  3 & & & 22 & \\
\end{array}
\]

\[
\begin{array}{c|cccc}
  i & j = 2 & 3 & 4 & \\
  1 & 33 & 23 & 31 & \\
  2 & 0 & 37 & & \\
  3 & & & 41 & \\
\end{array}
\]

b) After applying the modified sub-gradient algorithm:

(Take away 5 units from \( v = 5 \) and add 5 units to \( v = 6 \)) All of NewDMD1 is routed, and the residual NewDMD2 is:

\[
\begin{array}{c|cccc}
  i & j = 2 & 3 & 4 & \\
  1 & 0 & 0 & 0 & \\
  2 & 0 & 37 & & \\
  3 & & & 31 & \\
\end{array}
\]

This is the best that the algorithm can do. Given the physical network, and the unexpected demands, this is probably the optimal solution that could be theoretically obtained.

5. Summary and Discussion

In this letter, we have proposed a novel approach to integrating the physical layer planning with the logical layer configuration for multimedia ATM networks. The physical layer design takes advantage of ATM’s capabilities, and the logical layer configuration takes into account the capacities of the physical layer. The approach involves planning the initial physical layer; configuring/reconfiguring the logical layer; and then expanding the physical layer if necessary.

The objective function is also different from conventional ones in that a profit function (revenues minus costs) is maximized. Given the priorities of today’s network operators, this objective is more realistic than just minimizing costs. Within the objective function, two cost functions are developed that are more comprehensive than the ones in the literature so far.

Finally, we have developed a new heuristic algorithm that is simple to understand and use, and hence should be of practical value to network designers. The method quickly arrives at good (near-optimal) solutions, and should be scalable to realistically large networks. Thus ATM network optimization can be achieved by more designers and operators.

The solution method involves first designing a physical network by i] getting an initial feasible solution (IFS) and ii] improving on it iteratively. Next, the logical layer is configured by i] getting an initial routing scheme, and ii] improving on it by a method based on subgradient optimization. Finally, a method to expand the capacity of the physical layer is suggested.

We have illustrated the approach with some simple examples to demonstrate that the algorithms work and provide satisfactory results. We note that the main focus of this letter is on describing the approach, and other, more efficient algorithms may easily be incorporated into this approach to solve the optimization problems.

This letter proposes a new integrated model for ATM planning, but there are still many areas where further research is needed. Firstly, more efficient algorithms are needed to solve the model. It is important to find ways to increase revenues and decrease costs simultaneously so that the profit function is maximized. Secondly, this approach should be applied to solve for larger networks, and a greater variety of demand scenarios. Thirdly, we need to assess how good the results are by computing a good upper bound for the profit function. This is turn means computing good upper and lower bounds for the revenue and cost functions respectively. However, it may be noted that it will still be difficult to directly compare the results from this approach with conventional approaches because i] there are very few integrated models in the literature, and this model should only be compared to similar integrated models, and ii] the profit function and the cost functions are new. Fourthly, the algorithms should be extended to solve for the PL2 stage where the physical capacity of the network is expanded on the basis of shadow prices [7].
References