Virtual Path Topology Design, Control and Optimization*

Gang Feng, *Christos Douligeris and **Ramon M. Rodriguez Dagnino
Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33124-0640, USA
*University of Piraeus, Piraeus, 18534, Greece
**ITESM, Monterey, Mexico
Email: gfeng@ece.miami.edu, cdoulg@unipi.gr

ABSTRACT
An efficient virtual path (VP) control strategy is necessary in asynchronous transfer mode (ATM) networks to improve the utilization of network resources. Great advantages can be achieved by using the multistage VP control method, which is designed to execute the adjustment of VPs in three stages with specific tasks for each stage. In this paper, we first investigate the multistage VP control scheme, and then focus on the VP topology optimization (VPTO) problem which is of great importance in this method. An efficient objective function is constructed for this problem. A detailed analysis demonstrates that the objective function can reflect several performance parameters such as network throughput, processing costs and average packet delay. Moreover, a neural networks (NNs) approach is proposed to solve this problem to ensure real time implementation. Based on our recently presented stable state analysis technique, the NN system parameter values can be determined, thus enabling the NN to converge to a near optimal solution in each experiment. The validity and correctness of our algorithm are verified by extensive numerical results.

I. INTRODUCTION

In this paper, we hope to confirm the effectiveness of the multistage VP [1]-[2] control scheme implicitly or partially mentioned by some authors [4] – [8] and then provide a neural networks (NNs) solution to an optimization problem arising in this scheme.

The ultimate goal of VP control is to optimize the overall network performance by tuning the VP topology, i.e. adjusting the VP capacities and routes to the best state. Previously proposed VP control methods can roughly be classified into two categories, dynamic VP control and multistage VP control. Dynamic VP control tries to promptly adjust the VP topology to keep its fast response to the instantaneous changes of traffic flows. It seems that this method can dramatically improve the network throughput, but it may also suffer from an insurmountable drawback that the processing costs may considerably increase, which may even offset the advantages provided by the VP concept.

A VP control scheme should be able to mediate the contradiction between the throughput and processing costs. This includes some underlying requirements such that the capacity should be utilized efficiently and the VP topology should be kept in a good and also relatively stable state. Considering such features, this paper explores the multistage VP control method in which the adjustment of virtual paths is executed in three stages with specific tasks for each stage. In Section II, we first provide a brief description of this method and point out the primary goal of each stage, and then give more detailed discussions of each stage in subsequent Sections. As a critical issue in both the first and third stages, the VP topology optimization (VPTO) problem is elaborated next. The mathematical formulation of this problem with an objective function reflecting several performance parameters is given in Section IV. Numerical results are provided in Section V, while concluding remarks can be found in Section VI.

II. MULTISTAGE VP CONTROL SCHEME

The multistage VP control scheme includes three stages, which may be called static design, local modification and global reconfiguration, respectively. Each stage corresponds to a specific time scale for the VP control.

For a specific network, the first stage (static design) corresponds to the period of establishment or reset. The second stage (local modification) is in effect in the lifetime of the network, while the third stage (global modification) occurs at regular intervals or when the performance parameters being monitored exceed specified thresholds. In the first stage, the aim is to optimally design the VP topology based on the traffic parameters declared by users in advance on condition that the traffic requirements be satisfied. There is no restriction to the time to be consumed and the algorithms can be executed at the central node. The finally yielded VP information including the capacity, route and VPT pair of each VP is not only retained in the central node but also distributed to associated local nodes. In the second stage, the VP parameters are updated separately at each local node. Algorithms executed at a specific local node are limited to utilize the information of the virtual paths passing through this node, and that sent from the central node.

* This work has been supported in part by NSF under grant No. 9810534
and provided by the attached users. Adjustments, limited to those associated VPs, aim to track the instantaneous changes of traffic. Therefore, simple algorithms are preferred to guarantee their capabilities of fast response. The operations on VPs may include establishing new VPs, rerouting or tearing down existing VPs and modifying their capacities. The central node should be made aware of these local modifications since the global information of VP topology is needed to set up new VPs and also is necessary for the VPTO algorithm in the third stage. The VPTO should result in maximized network performance, which may have been aggravated after a period of local operations.

The salient features of the multistage VP control method are its feasibility and flexibility. Since it is unnecessary for local nodes to interchange information and the responsibility for VP control is shared by the central node, the processing load at local nodes may be alleviated and well-balanced. The amount of information interchanged between the central node and local nodes can be limited to an acceptable range by adjusting the threshold that triggers the interchange. Furthermore, the stability of VP topology can be maintained with satisfactory network performance as long as an appropriate updating interval is chosen.

III. STATIC DESIGN OF VP TOPOLOGY

The static design of VP topology can be described as the following optimization problem: Given a physical topology, capacities of physical links and traffic requirements of source-destination (SD) pairs, determine the VPT pair, physical route and capacity of each VP with the objective of optimizing the overall network performance. This is only a qualitative description. Different implementers may adopt distinct objective functions and decision variables.

A. Two-Phase Static Design

Much work has been done for the static design problem [4]–[8]. Based on previous research results, we herein present a two-phase design method for the aforementioned VP topology static design problem. The goal in the first phase is to determine VP terminators, capacities and a set of alternate routes for each VP, while in the second phase we try to choose a route for each VP from the corresponding set of alternate routes so that the overall network performance is optimized.

The problem setting in the second phase is the same as that in the third stage of the multistage VP control as mentioned in Section II, and correspondingly can be formulated as the same VP topology optimization problem, which will be discussed later. In this Section, we focus on the first phase. Some considerations on the related problems are given as follows.

(1) Determining the terminators and quantity of VPs

The determination of terminators and quantity of VPs is highly related to the limited number of bits allocated to the VPI field in the ATM cell header. For a network of arbitrary size, the ATM cell has the same format, in which a VPI value is always restricted to 8 bits in a user-network interface (UNI) cell header and 12 bits in a network-network interface (NNI) cell header. For this reason, the amount of VPs seems to be excessive in some cases, while inadequate in some other cases.

For a network of relatively small size, we may consider all the nodes directly connected to users as VP terminators, and by doing so we are able to form a fully meshed VP network containing only direct VPs. Moreover, it is possible that more VPs may be established between any SD pair as long as enough VPs are available. For instance, we may assign a VP to transmit each service so that the QoS can be strictly controlled.

For relatively large-scale networks, we can reduce the number of VPs so that the VP network can remain fully interconnected only by direct VPs. Based on the suggestions of Liu et al. [9], two methods can be used to reduce the number of VPs. First, between any SD pair, we may let one VP to transmit all CBR and VBR services, while another VP to transmit all ABR and UBR services. Second, we may design a new VP switch which can identify the physical link port ID (PID). An ATM cell incident to a switch has to determine its outgoing route in terms of both VPI and PID.

For very large-scale networks, or when there is difficulty in realizing VP switches capable for PID identification, we may divide some direct VPs into two serially connected VPs. In this case, some switches in the core network must be set to be VP terminators.

(2) Determining the VP capacities

At the static design stage we may set an initial value, which is not necessarily the best one, for the capacity of each VP. This consideration is based on that we can perform further adjustments on the VP capacities after the second phase of the static design. Moreover, in the second stage of the multistage VP control some algorithms can be employed to perform finer tuning on the capacities so that they can be best utilized.

More precisely, for the VP transmitting CBR/VBR services the peak rate declared by the user can be taken to be its capacity, while for the VP transmitting ABR/UBR services the lowest rate declared can be taken to be its capacity. Other methods such as the calculation of effective bandwidth [3] are also applicable. After the second phase of the static design, the capacity of each VP can be enlarged proportionally as long as the total amount of bandwidth accommodated by a link does not exceed its physical capacity.

(3) Determining the alternate routes
In order to establish a VP between a SD pair, what physical route should be chosen to accommodate it? A stringent mathematical analysis of this problem involves the need to solve an NP-hard combinatorial optimization problem. However, for a specific VP, if we make a manual choice of its route, most routes traversing a long distance between the SD pair will definitely be excluded from consideration, and only those approximately “shortest” routes will be considered as candidates. The alternate routes between a SD pair can be obtained using the K-shortest path algorithm or the recursive algorithm provided in [11].

IV. GLOBAL RECONFIGURATION: VP TOPOLOGY OPTIMIZATION

Global reconfiguration can be formulated as an optimization problem, which is based on the following conditions: the physical topology, the available capacity of each link, VP terminators, and the bandwidth and a set of alternate routes for each VP.

A. Objective Function

In this paper, our final goal is to maximize the overall network throughput, which can be decomposed into four objectives. Two of them are related to the link capacity utilization ratio (the ratio of the offered capacity to the total capacity for a specific link): the minimization of the summation of all link capacity utilization ratios, and the minimization of the square summation of all link capacity utilization ratios. The other two are similar ones associated with the nodal capacity utilization ratio (the ratio of the amount of information traversing the node to the maximal throughput of the node): the minimization of the summation of all nodal capacity utilization ratios, and the minimization of the square summation of all nodal capacity utilization ratios.

In order to represent these objectives in a mathematical formulation, we define the following notations. Assume that an ATM network is represented by a directional graph \( G(V, E) \), where \( V \) is the set of nodes, while \( E \) is the set of links. Assume there are a total of \( K \) physical links, the capacity of the \( k \)th link is \( C_k \) (bits/s), and the amount of information traversing the link is \( f_k \) (bits/s). Similarly, we assume there are \( S \) nodes in total and \( C_k^+ \) and \( f_k^+ \) are the maximal capacity and offered capacity of the \( k \)th node, respectively. By means of these notations, we can express the above four objectives as follows:

\[
\text{Minimize } \sum_{k=1}^{K+S} \left( \beta_k u_k + u_k^2 \right), \tag{1}
\]

where

\[
u_k = \begin{cases} \frac{f_k}{C_k}, & k \in [1, K] \\ \frac{f_k}{C_k^+}, & k \in [K+1, K+S] \end{cases}, \quad \text{and } \beta_k \text{ is a constant.}
\]

In [12], we showed if \( \beta_k = \frac{1}{1-\mu_k} \), our objective function would be equivalent to the expression of the average packet delay [13]. This indicates that, to a certain extent, the objective of maximizing the throughput is equivalent to minimizing the average packet delay.

B. VP Topology Optimization Based on Neural Networks

In this subsection we propose a method based on the HNN [14] to solve the VP topology optimization problem. The HNN is a single-layer feedback network interconnected by a large number of neurons. When the network converges from a random initial state to the final stable state, the vector of outputs of neurons corresponds to a minimum point in the hyper-cube, which is reflected by the fact that the associated Lyapunov energy function falls into a local minimum. The neuron’s input-output relation is generally characterized by a sigmoid function (which is also used in this paper),

\[
V_i = \frac{1}{1 + e^{-\frac{I_i}{\theta}}}, \tag{2}
\]

where \( U_0 \) is a constant characterizing the nonlinear slope, \( V_i \) and \( U_i \) are the output and input voltages of neuron \( ij \). The key issue in the use of the HNN to solve an optimization problem is how to construct an energy function that can best reflect its objectives and constraints.

In order to construct an energy function for the VP topology optimization problem, we need to define the following notation:

\[
M \quad \text{total number of VPs} \\
R_i \quad \text{number of alternate routes for the } i\text{th VP} \\
B_i \quad \text{bandwidth of the } i\text{th VP (bits/s)} \\
L_{ijk} \quad \text{equals to 1 if the } j\text{th route of the } i\text{th VP goes through the } k\text{th link and 0 otherwise} \\
\]

The amount of information passing through the \( k \)th link (or node) can be expressed as follows:

\[
f_k = \sum_{i=1}^{K} \sum_{j=1}^{R_i} B_i V_i L_{ijk}, \quad k=1, \ldots, K+S. \tag{3}
\]
where \( L_{ik} = \begin{cases} L_{q_k}, & k \in [1, K] \\ \sum \forall j, & k \in [K + 1, K + S] \end{cases} \).

We let each alternate route correspond to a neuron, and assume that the \( i \)th alternate route of the \( i \)th VP is finally determined to accommodate it if at the stable state the output of neuron \( ij \), i.e. \( V_{ij} \), equals to 1.

Now consider what terms should be included in the energy function. The only constraint in the VP topology optimization problem is that for each VP only one alternate route is finally chosen. This can be transformed as the following two energy terms:

\[
E_i = \sum_{j \neq 1} (\sum_{s \neq j} V_{ij})^2,
\]

and

\[
E_2 = \sum_{i \neq 1} \left( \sum_{j \neq 1} V_{ij} - 1 \right)^2.
\]

In order to include the objective function (1) into the energy function, we need to use a constant \( \beta \) to substitute \( \beta_k \) due to the reason that the Hopfield energy function must be of a quadratic form, whereas \( \beta_k \) involves a division computation. Thus the energy term reflecting the objective function is given by

\[
G = \sum_{i \neq 1} \left( u_i + \beta \alpha_i^2 \right).
\]

Eventually the energy function can be expressed as a weighted sum of the above three terms:

\[
E = AE_i + \frac{B}{2} E_2 + CG.
\]

where \( A, B \) and \( C \) are positive constants.

Taking a partial derivative of (7) with respect to \( V_{ij} \) gives rise to the following motion equations:

\[
\frac{dU_{ij}}{dt} = -\frac{\partial E}{\partial V_{ij}} = -A \sum_{s \neq 1} V_{is} - B \left( \sum_{s \neq 1} V_{is} - 1 \right) - CG_y.
\]

where \( g_y = \frac{\partial G}{\partial V_{ij}} = \sum_{s \neq 1} (1 + 2 \beta \alpha_i^2) \frac{B_{ij}}{C_s} \).

Parameters \( A, B \) and \( C \) have a great influence on the quality of the achieved solutions. A stringent analysis for the choice of reasonable values is very complicated. Therefore, we only provide the result obtained by means of the stable state analysis technique proposed in [10]. The following two inequalities should be satisfied when choosing the values of \( A, B \) and \( C \):

\[
B > C \cdot \text{Max}(g_y, \forall i, j),
\]

where \( \text{Max}() \) returns the maximal value, and

\[
A < C \cdot \text{Min} |g_y - g_{ij}| \forall i, j, k, i \neq k.
\]

where \( \text{Min}() \) returns the minimal value, \( |\cdot| \) denotes the absolute value operator.

V. EXPERIMENTAL RESULTS

A. Network Model

In this Section, the proposed HNN-based VPTO algorithm is applied to a network to test its performance. The network model, which is same as the one analyzed in [15], includes 10 nodes and 36 links (or 18 duplex connections), as shown in Fig. 1. Node 9 is a geosynchronous satellite. The capacities of each terrestrial link and each of the six satellite links are 38.4 Kbps and 50 Kbps, respectively. Besides, we assume that the maximal throughput of each node is 150 Kbps, and 29 VPs are to be established in this network. The source/destination node and the traffic requirement of each VP are enumerated in Table I.

![Fig. 1. The network model](image)

B. Alternate Routes and Parameter Values

By means of the algorithm proposed in [11], we obtain 4 alternate routes for each VP (not shown here due to limitation of space).

The computer simulations are based on the discrete form of (8). In all of our experiments, the time step-size is set to 0.0001, and the iteration number is 1000. The parameter \( U_{ij} \) in the sigmoid function is set to 0.01. The values for parameters \( \beta, A, B, C \) are as follows: \( \beta=100, A=1, B=50, \) and \( C=1 \).

The quality of the corresponding solution is investigated in terms of the time delay, which is calculated by

\[
T = \lambda \sum_{i \neq 1} \frac{u_i}{1 - u_i}.
\]

C. Analysis and Comparison of Experimental Results

We have done 100 experiments continuously. In each experiment, the HNN can always converge to a valid solution starting from a random initial state (the initial
output of each neuron is set to a deterministic value 0.25 plus a random number in the interval [0.02, 0.02]. Among the 100 results, the maximum is 0.040385 (s), the minimum is 0.040153 (s), and the average is 0.040246 (s).

We have also tested the network model by means of the shortest path algorithm and the least loaded algorithm in terms of the following objective functions [8]:

\[
\min_j \left( H_{ij} \right) \tag{12}
\]

\[
\max_k \left( \min_{i \in P_k} \left( \text{capacity}(k) \right) \right) \tag{13}
\]

and

\[
\max_j \left( \frac{1}{H_{ij}} \sum_{k \in P_j} \text{capacity}(k) \right) \tag{14}
\]

where \( P_{ij} \) is composed of all the serial numbers of the links traversed by the \( j \)th alternate route of the \( i \)th VP, \( H_{ij} \) is the number of nodes traversed by route \( P_{ij} \), and function \( \text{capacity}(k) \) returns the residual capacity of the \( k \)th link. The achieved average packet delays corresponding to the three objective functions are 0.069056 (s), 0.067671 (s), and 0.042821 (s), respectively. Apparently, each of the three results is much worse than any one of our results.

VI. CONCLUSIONS

In this paper, the multistage VP control scheme is elaborated. A detailed analysis demonstrates that this scheme has many advantages such as feasibility and flexibility. We also provide an HNN-based approach to solve the VP topology optimization problem arising in the static design and global reconfiguration stages of this scheme. Since the hardware-implemented HNN can produce a solution in a very short time, the proposed algorithm is likely to meet the requirement of responsiveness for the global reconfiguration. Computer simulations also indicate that the proposed algorithm can generate much better solutions than other proposed methods.

REFERENCES


<p>| TABLE I: TRAFFIC REQUIREMENTS OF VPS (CELL/S) |</p>
<table>
<thead>
<tr>
<th>VP #</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Destination</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Traffics</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>14</td>
<td>13</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>VP #</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Traffics</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>