A Design of Self-healing ATM Networks Based on Backup Virtual Paths

Chae Y. Lee and Seok J. Koh

Department of Industrial Engineering, KAIST

373-1, Kusung-Dong, Taejon, 305-701, Korea

E-mail : cylee@ie1.kaist.ac.kr

Scope and Purpose

Broadband network will support a variety of services such as voice, video, data and image with different traffic characteristics and different qualities of service in integrated environment. Asynchronous Transfer Mode (ATM) network techniques have been actively studied with the goal of realizing B-ISDN. ATM network will provide the infrastructure for B-ISDN and offer the benefit of enhanced flexibility when the virtual path (VP) concept is adopted. One of the most important issues in B-ISDN is the increased level of network survivability. The backup VP scheme prepares a preassigned backup VP before a failure occurs. A backup VP can be assigned between any pair of nodes on a VP route. The backup route is link disjoint from the original path to avoid the simultaneous failure of both the original and the backup VP. In this paper, a self-healing VP network design is considered to find the configuration of the backup VP such that the cost of required spare capacity is minimized.

Abstracts

We propose a design of self-healing ATM virtual path (VP) networks based on backup VP. The backup VP is a preassigned virtual path to restore the traffic after a network failure. It is known that the self-healing scheme based on the backup VP is simple, fast and easy to implement. Given a set of bandwidth requirements and original routes for target demands, the design is to find the backup route of each traffic demand such that the cost of total spare bandwidth, which is required for each physical link to restore the services, is minimized. The problem is formulated as an integer linear program. Based on the property that a common backup VP of any two distinct traffic demands saves the required bandwidth, an initial and two improvement heuristics are proposed. Computational experiments show that the proposed procedure provides solutions the gap of which are less than 11% compared to optimal solutions.

I. Introduction

Broadband network will support a variety of services such as voice, video, data and image with different traffic characteristics and different qualities of service in integrated environment. Asynchronous Transfer Mode (ATM) network techniques have been actively studied with the goal of realizing B-ISDN. ATM network will provide the infrastructure for B-ISDN and offer the benefit of enhanced flexibility when the virtual path (VP) concept is adopted.

In ATM networks, the VP concept for path layer has been standardized in ITU-T [20]. A VP is a labeled path established by the VP identifier (VPI) and VPI tables in cross-connect node. The VPI is a number contained in the cell header that identifies the assigned path of the cell. The VP is different from the digital path in Synchronous Transfer Mode (STM) network. The route and bandwidth of a VP are defined independently. The route is defined in the VPI table of the cross-connect nodes, while the bandwidth is logically defined and managed by the VP terminal equipment in ATM switching offices. By this independence of route and bandwidth establishment, a VP route can be established without assigning its bandwidth along the path. In other words, it is possible to assign zero bandwidth to a VP. In this paper, zero bandwidth VP is used as a backup VP for restoration of a failed VP. When a VP fails, its backup VP is activated by capturing the required bandwidth out of the spare capacity of physical links [5,6,7].

One of the most important issues in B-ISDN is the increased level of network survivability. In B-ISDN, the volume of carried traffic greatly increases with the economic activity it must support. In particular, the bulk of the traffic carried will be data of high information value. Therefore, the damage by a single transmission facility failure becomes much more serious than in existing networks. Thus it is very important to develop rapid restoration scheme to minimize the economic and social damage.

There have been a lot of studies on the self-healing architectures for STM network such as the physical link topology and architecture design with the self-healing ring [4,9,11,12,14,18], while the self-healing schemes for ATM network has not been studied enough. Self-healing implies the restoration of failed paths with distributed control mechanism. Two self-healing schemes for ATM networks have been recently proposed based on flooding [10,15] and backup VP mechanisms [1,5,6,7].

The ATM flooding algorithm works as follows [10]. If a link fails, the downstream node of the failed link becomes a sender node while the upstream node becomes a chooser node. Subsequently the sender node starts broadcasting restoration messages for all VPs on the failed link via every possible alternate routes to the chooser node. The restoration messages are rebroadcasted at intermediate nodes on links, and the VP routing tables are updated. The flooding scheme searches for restoration routes by broadcasting messages after a failure is detected.

The backup VP scheme, however, prepares a preassigned backup VP before a failure occurs [5,6,7]. Figure 1(a) illustrates the self-healing scheme for ATM network using a backup VP. Initially, the backup

route for the original VP is preassigned and the bandwidth of the backup VP is set to be zero. After a failure is detected, VP-AIS (Alarm Indicate Signal) cells are generated and transmitted as soon as possible in order to indicate an interruption of cell transfer. By using OAM (Operation, Administration and Management) cells [21], the node then sends a restoration message including required VP bandwidth along the backup VP. Path restoration is realized by capturing the bandwidth of the backup VP from shared spare capacity, and redirecting cells from the failed VP to the backup VP. A backup VP can be assigned between any pair of nodes on a VP route. The backup route is link disjoint from the original path to avoid the simultaneous failure of both the original and the backup VP.

It is known that the backup VP scheme is faster and simpler and easier to implement than the flooding scheme [7]. However, effective ATM network design based on the backup VP scheme has not been studied enough. The goal in the network design is to minimize the spare capacity of physical links which is required for VP restoration. In this paper, a self-healing VP network design is considered to find the configuration of the backup VP such that the cost of required spare capacity is minimized. In particular, multiple backup VP's are employed for the original VP to improve the flexibility against the unpredictable change of network environments. Note that the congestion by irregular traffic fluctuations or double link failures may make it impossible to restore the original VP through single backup VP. Such an inflexibility can be improved by providing two different backup routes for the original VP. Moreover, we can effectively assign several backup VPs to a high-priority VP to improve reliability against multiple link failures. Figure 1(b) illustrates the restoration algorithm of the double backup VP scheme. When a failure occurs, the downstream node of the restoration pair detects VP failure by receiving VP-AIS cells. By using OAM cells, the node then sends a restoration message including required VP bandwidth along the backup VP. If the backup VP is not usable because of the congestion or another failure, then "uncapturable message" returns to the downstream node and alternate backup route (the secondary backup VP in the figure) is employed to restore the failed VP. When the upstream node of the restoration pair receives the restoration message, it switches traffic from the failed VP to the backup VP. This completes failure restoration for the VP.

In the dimensioning of ATM VP network [3,8,13,17], the route and bandwidth requirement of the original VP are obtained. In the backup VP scheme considered in this paper, the route of each backup VP and spare resource in each link are determined to minimize the cost of network resources required. Particularly, an ATM network design is considered to find the route of backup VP that is link disjoint from its original VP such that the cost of total spare capacity required is minimized.

This paper is organized as follows. A network design based on the backup VP scheme is proposed as an integer programming model in Section II. In Section III, an initial and two improvement algorithms are developed to solve the problem. The effectiveness of the proposed algorithms is demonstrated with computational experiments in Section IV. Finally, Section V concludes this paper.

II. A Model for the ATM Network Design

In this section, a network design problem is described to find routes of the backup VP for each original working VP such that the cost of spare capacity required is minimized. The problem is modeled as an integer linear program.

In the network, the route and the bandwidth requirement for each working VP are given by solving the dimensioning problem [3,8,13,17]. The dimensioning can be decoupled into two subproblems; bandwidth estimation and VP link sizing problem [13]. Given a set of Quality of Service (QoS) traffic requirements such as call blocking, cell delay and cell loss probability, the bandwidth requirement is estimated as an equivalent bit rate (EBR) for each VP [2]. To utilize the statistical multiplexing gain, EBR is calculated based on the traffic descriptors; traffic load, peak rate, mean burst period and buffer overflow probability. In VP link sizing, the capacity of each physical link is determined by finding the route of the working VP such that the bandwidth requirement is satisfied for each VP.

In this paper, it is assumed that the bandwidth requirement and the route of working VP for each target demand are given in the dimensioning problem. The problem considered is to find single or double backup routes that should be link disjoint from the working route for each target VP. Under normal operations, no bandwidth is assigned to each backup VP. When a link fails, however, the backup VPs become active to restore the target VPs which are affected by the failed link. The objective is to minimize the cost of total spare capacity required for VP restoration.

The network $G(N,A)$ to be considered is composed of a set of ATM switches and/or cross-connects N and a set of directed links A. Let us denote a set of undirected links by E, that is, each undirectional link in E consists of two directional links in A. We introduce K to represent a set of demands or target VPs and $O(k)$ and $D(k)$ to denote, respectively, the origin and the destination node of a demand $k \in K$. Let us denote $L(k)$ by a set of directed links which the working VP of the demand k traverses in the network and $VP(e) \subseteq K$ by a set of demands which are affected by a failure of an undirected link $e \in E$. Note that $VP(e)$ can be obtained by finding the demands, the working route of which traverses the link e. For mathematical formulation of the problem, let us define the following notations.

c_{ij}: unit bandwidth cost of a directed link (i,j).

y_{ij}: an amount of the spare bandwidth required for a directed link (i,j).

x_{ij}^k: 1 if the backup VP of demand k is routed through the directed link (i,j), 0 otherwise.

r^k: an integer number of backup VPs required for the demand k. (e.g. 1 or 2)

b^k: the bandwidth requirement for a demand k. (e.g. Mbps)

α: unit capacity of spare bandwidth (e.g. 1.5 Mbps = T1 carrier rate).

Note that each demand requirement b^k is transmitted over an integer number of T1 carriers whose unit capacity is α. Then we propose the following model for designing a self-healing ATM network based on the backup VP. The problem is to find the route of backup VP which is link disjoint from its working VP, such that the cost of total spare capacity required is minimized.

$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{j:(i,j)\notin L(k)} x_{ij}^k - \sum_{j:(i,j)\notin L(k)} x_{ji}^k = \begin{cases} r^k & \text{if } i = O(k) \\ -r^k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } k \in K \tag{2}$$

$$\sum_{k\in VP(e)} b^k x_{ij}^k \leq \alpha y_{ij} \qquad \text{for all } (i,j) \in A, e \in E \tag{3}$$

$$x_{ij}^k = \quad 0 \text{ or } 1 \qquad \text{for all } (i,j) \in A, k \in K \tag{4}$$

$$y_{ij} \geq \quad 0 \text{ and integer} \qquad \text{for all } (i,j) \in A \tag{5}$$

In the model, the objective function (1) represents the cost of total spare bandwidth for network links. The link cost is represented as the sum of several cost components; terminal cost, fiber materials costs, fiber placement and repeater cost. The terminal cost is the working and protection equipment cost for termination including an ATM switch at both ends of each fiber system in the link. The terminal cost is a function of the number of T1 carriers and fiber transmission rate. Fiber material cost depends on the number of fiber systems used and the length of each fiber system. The fiber placement cost is the installation cost of fiber cables. A repeater is required if the length of the link exceeds a distance threshold. In the network planning period, it is very difficult to estimate each fiber cost components exactly, since fiber technologies will be rapidly improved. However, total link costs are approximately proportional to the link capacity [13,18]. Thus, in this paper, the linear link cost assumption is made.

In constraints (2) and (4), r^k backup routes are selected between the origin and the destination node of each target demand k such that one backup route is link disjoint from its working route as well as the other backup route. In (2) the backup route does not pass through the link (i,j) in $L(k)$. The right hand side of (3) represents the amount of spare bandwidth required for a physical link (i,j) against a failure of any link e. When a physical link e fails, each working VP that passes through this link also fails. Thus several backup VPs may become active simultaneously when a link fails. In (3) and (5), y_{ij} is the number of basic units of capacity 1.5 Mbps which is required for the link (i,j). Note that the solution of the model provides the spare capacity required as well as the route of each backup VP. In case of $r^k = 2$, the model guarantees that each target demand can be restored by its secondary backup VP even if its working and primary backup VPs fail simultaneously.

In the model proposed in this paper, it is assumed that the working route for each target demand is given. Note that there is another version of the model in which the working routes and the backup routes are determined simultaneously. In such a model, under normal network operation, working routes can be reset in order to provide better quality of service to network users by using the spare link capacities. In addition, when a link fails, working routes which are not affected by the link failure can be also rerouted to utilize spare link capacities more efficiently. However, those models are not considered in this paper.

III. Algorithms for the Design of ATM Backup Routes

In this section, algorithms are developed to solve the self-healing ATM network design problem. We first examine the existing studies by Kawamura *et. al.* [6]. Next, structural properties of the problem are developed. An initial heuristic is developed to obtain feasible routes of backup VPs. Two improvement heuristics are also proposed based on the structural properties.

A. *Alternate Algorithm by Kawamura et. al. [6]*

For the network design with single backup VP, a simple heuristic has been proposed by Kawamura et. al. [6]. In the heuristic, an initial backup VP route is obtained by finding the shortest path which is link disjoint from the working route. The algorithm "Alternate" is summarized as follows.

Alternate Algorithm

Step 1. Set an initial backup VP routes as the "shortest path" for each target demand $k = 1, .., |K|$.

Step 2. Consider a target demand pair $k = 1$.

Step 3. Search all alternate backup VP routes for the demand k.

Step 4. For each alternate route, calculate the amount of required spare capacity on each link.

Step 5. From Step 4, if the network cost is improved, then the alternate route is newly adopted.

Step 6. Until all alternate routes are checked for the demand, do Step 4 - 5.

Step 7. Set k=k+1 and go to Step 3.

Step 8. Until all target demands are considered, do Step 3 - 7.

Step 9. Until no further cost improvement is made, do Step 2 - 8.

In Step 4, the amount of required spare capacity is computed as follows (see the Appendix of [6]). For each alternate backup VP route obtained from Step 3, "in order to guarantee complete restoration against any node failure, the spare capacity $S(l)$ on each link l must satisfy the following condition; $S(l)$ >= max $\{x| x = C(l,n)$ for all $n\}$, where $C(l,n)$ is the required spare capacity on link l when a node n fails". In other words, any demand whose working VP becomes active due to the failure of node n requires its capacity on link l of the backup VP. $C(l,n)$ thus represents the sum of all such demand capacities. Finally,

the maximum required capacity at link l is determined by $S(l)$ whatever the failed node is. Note that the network cost in the algorithm is calculated from $S(l)$. In Step 5, if the network cost is improved by using the alternate route, the alternate backup VP route is newly adopted in the network.

B. Common Backup Path of Multiple Demands

In this section we present properties which are useful in minimizing the cost of backup VPs. The properties are employed in obtaining initial and improvement heuristics.

We assume that each demand has two VPs; one for working and the other for backup. Now, consider two working VPs of two different demands. By assuming that the two working VPs are link disjoint, they cannot fail simultaneously for any single link failure. Thus the backup VPs for the two working VPs cannot become active simultaneously. Accordingly, the network cost may be reduced by routing those two backup VPs through a common path. The spare capacity required for each link of the common backup path is determined by the maximum value of the two demand requirements.

Let WP(k) and BP(k) be respectively the working and the backup routes for a target VP of demand k. Consider WP(k^*) and BP(k^*) for another VP of demand k^* such that WP(k^*) is link disjoint from WP(k). Given the backup route BP(k), it is cost-effective to restore the BP(k^*) through the links in BP(k), since two backup routes BP(k) and BP(k^*) cannot be active simultaneously for any single link failure. These statements are summarized in Property 1 and illustrated in Figure 2.

Property 1. Given the working routes WP(k) and WP(k^*) that are link disjoint each other, we assume that the backup route BP(k) is given with a spare bandwidth b^k. Then each link in the backup route BP(k^*), which is shared by BP(k), requires the following additional bandwidth;

   Case 1. If $b^k \geq b^{k^*}$ then no additional bandwidth is required.
   Case 2. If $b^k < b^{k^*}$, then $(b^{k^*} - b^k)$ is required additionally.

Now let us consider more than two demands. Let WP(k), $k \in K$ be the working routes and BP(k), $k \in K$ be the backup routes. Consider WP(k^*) which is link disjoint from each WP(k). Then Property 1 is generalized as in Property 2.

Property 2. Given the working routes WP(k), $k \in K$, suppose that WP(k^*) is link disjoint from WP(k). We assume that each backup route BP(k) is given with a spare bandwidth b^k. Then each link in the backup route BP(k^*), which is shared by BP(k) requires the following additional bandwidth;

   Case 1. If $\max_{k \in K} b^k \geq b^{k^*}$, then no additional bandwidth is required.
   Case 2. If $\max_{k \in K} b^k < b^{k^*}$, then $(b^{k^*} - \max_{k \in K} b^k)$ is required additionally.

   These two properties imply that the network cost can be improved by routing several backup VPs

7

such that they share a common route. Based on the properties described above, an initial and two improvement heuristics are proposed in the following two sections. Each heuristic first check if two target VPs are link disjoint or not. If those target VPs are link disjoint in the network, the network cost is reduced by routing those corresponding two backup VPs through a common path.

C. Initial Heuristic

One of the simplest procedure to get an initial feasible solution of the problem is to find the shortest backup VP route for each demand pair. In this section, we propose a new initial heuristic which uses a modified link cost to find a feasible backup VP route based on the properties introduced in the previous section.

Given the configuration of the working routes and bandwidth requirements, an algorithm is proposed to get initial feasible routes of the backup VPs in which each backup route is link disjoint from the working route. We assume that demands are numbered such that $b^1 \geq b^2 \geq ... \geq b^D$, where $D = |K|$. In Case 1 of Property 1, if $b^k \geq b^{k*}$ then no additional bandwidth for the backup links of the demand $k*$ is required. For $k = 1$, we find the shortest path that is link disjoint from its working route. If $r^k = 2$, we also find the shortest path after deleting the links on the backup routes that have already been obtained.

Now for $k = m$, the target VP of demand m is considered, by assuming that each backup route has been obtained for demands $k = 1, ..., m$-1. Among the demands $k = 1, ..., m$-1, we find a set of demands such that WP(k) is link disjoint from WP(m). We define $K(m)$ as the demand set such that WP(k) is link disjoint from WP(m) among the demands $k = 1, ..., m$-1. Note that $K(m)$ may be empty. Otherwise, it is clear that $b^k \geq b^m$, since each demand is sorted in the nonincreasing order. Then we find a set of links $A(m)$ which is the set of links on the backup routes of demands in $K(m)$. By Property 1 and 2, each link in $A(m)$ does not require any additional bandwidth even if the backup VP of demand m is routed through the link. Thus each link cost c_{ij} is updated as follows. Set $c_{ij}* = 0$ for all (i,j) $\in A(m)$ and $c_{ij}* = c_{ij}$, otherwise. Based on the new link cost $c_{ij}*$, we find the r^m shortest paths that are link disjoint from the working route. These steps are continued until all the backup VPs are obtained.

Working VP Based Algorithm (WVPBA)

Step 1. Set $k = 1$.

Step 2. Find the r^k shortest paths that are link disjoint from WP(k).

Step 3. Set $k = k +1$.

   Step 3-1. Find $K(k)$ and $A(k)$ for the demand k.

   Step 3-2. If $K(k)$ and $A(k)$ are not empty, modify each link cost as $c_{ij}*$ for all (i,j) $\in A(k)$

   Step 3-3. Find the r^k shortest paths that are link disjoint from WP(k) based on $c_{ij}*$.

Step 4. If $k < |K|$, go to Step 3. Else if $k = |K|$, go to Step 5.

Step 5. Compute the network cost and stop.

In Step 5, the network cost is computed by summing the link cost of each link $(i,j) \in A$. Given the configuration of the backup routes, the cost of link (i,j) is determined by $c_{ij} y_{ij}$, where $y_{ij} = \max_{e \in E} \left\lceil \sum_{k \in VP(e)} b^k x_{ij}^k / \alpha \right\rceil$ from Equation (3) of the proposed model.

D. Two Improvement Heuristics

In this section two new improvement heuristics; the backup VP based algorithm (BVPBA) and the maximum link based algorithm (MLBA) are developed based on the structural properties described in Section III-B. Given the working and the backup routes obtained in the initial heuristic algorithm, the improvement is performed by alternate backup routes the cost of which is computed by the change of bandwidth explained in Properties 1 and 2. Another algorithm improves the initial solution by reducing the maximum link cost of the network.

The BVPBA is based on the Alternate heuristic [6]. However, Alternate heuristic does not use any information of network structure, while the BVPBA finds alternate backup paths on the basis of the modified link cost. Given an initial solution x^k for each demand k, the backup VP cost is determined by $\left\lceil \sum_{(i,j) \in A} c_{ij} b^k x_{ij}^k / \alpha \right\rceil$ on the model in Section II. The backup VP cost represents the cost of total spare bandwidth required to activate the backup VP for demand k. After computing the backup VP cost of each demand, we sort the demands in nonincreasing order of the backup VP cost. The BVPBA improves the network cost by reducing the maximum backup VP cost.

Let us consider a target demand m. We define $K(m)$ as the demand set such that WP(k) is link disjoint from WP(m) for $k = 1, ..., |K|$. Then we find a set of links $A(m)$ which is the set of links on the backup routes of demands in $K(m)$. By Property 2, each link in $A(m)$ requires relatively small additional bandwidth even if the backup VP of demand m is routed through the links. Thus each link cost c_{ij} is updated as follows. For all $(i,j) \in A(m)$, if $\max_{k \in K(m)} b^k \geq b^m$, then set $c_{ij}^* = 0$, else set $c_{ij}^* = c_{ij} (b^m - \max_{k \in K(m)} b^k)/b^m$. For all $(i,j) \notin A(m)$, $c_{ij}^* = c_{ij}$.

Based on the modified link cost, the network cost is improved as follows. For each backup VP, find the w alternate shortest routes that are link disjoint from its working route, where w is an integer number so that w alternate routes are examined to improve the network cost. Note that each alternate backup route should be link disjoint from its working route as well as the other backup route of the target demand. For each alternate route, if the route improves the network cost, it is newly adopted. These steps are repeated for each backup VP until no further cost improvement is obtained. The BVPBA is summarized as follows.

9

Backup VP Based Algorithm (BVPBA)

Step 1. Sort all the backup VPs in the nonincreasing order of the backup VP cost $\lceil \sum_{(i,j) \in A} c_{ij} b^k x_{ij}^k / \alpha \rceil$.

Step 2. Let N be the number of all backup VPs considered in the network. Set $n = 1$.

Step 3. For the backup VP n, do the following steps;

Step 3-1. Find $K(m)$ and $A(m)$ for demand m, where m is the demand index of the backup VP n.

Step 3-2. If $K(m)$ and $A(m)$ are not empty, modify each link cost as c_{ij}^* for all $(i,j) \in A(m)$

Step 3-3. Find the w alternate shortest routes that are link disjoint from WP(m) based on c_{ij}^*.

Step 3-4. Do the followings until all the w alternate routes are examined.

If the alternate route improves the network cost, it is newly adopted and go to Step 4.

Step 4. If $n < |N|$, then set $n = n + 1$ and go to Step 3. Else go to Step 5.

Step 5. If the network cost is improved in Step 3 and 4, then go to Step 1. Else stop.

The MLBA is different from BVPBA and Alternate heuristic in the sense that the solution is improved by reducing the maximum link cost of the network. To reduce the maximum link cost of the initial solution, we first sort all the links in the nonincreasing order of the link cost $c_{ij} y_{ij}$ obtained from the initial backup routes. Let (a,b) be the link with the maximum link cost in the network. The network cost is improved by rerouting the backup routes, which pass through the link (a,b), to the other alternate route.

For each backup VP traversing the link (a,b), we obtain the demand set K, the link set A and the modified link cost c_{ij}^* as in Step 3-1 and 3-2 of the BVPBA algorithm. Then we find w alternate shortest routes satisfying the following route condition; the alternate backup route is link disjoint from the working route and should not pass through the link under consideration. For each alternate route, if it improves the network cost, it is newly adopted. Repeat these steps for all backup VPs which pass through the link (a,b). Note in the algorithm that all the links are examined to improve the network cost in the order of nonincreasing link cost, until no further cost improvement is obtained. The MLBA is summarized as follows.

Maximum Link Based Algorithm (MLBA)

Step 1. Sort all the links in the nonincreasing order of link cost $c_{ij} y_{ij}$..

Step 2. Let L be the number of the links in the network. Set $l = 1$.

Step 3. For the link l, do the following steps;

Step 3-1. Let N be the number of backup routes which pass through the link l.

Step 3-2. For the backup route $n = 1$ to N, do the following steps.

A) Find $K(m)$ and $A(m)$ as in BVPBA. where m is the demand index of the backup route n.

B) If $K(m)$ and $A(m)$ are not empty, modify each link cost as c_{ij}^* for all $(i,j) \in A(m)$

C) Find the w alternate shortest satisfying the route condition with a link cost $c_{ij}*$.

D) Do the followings until all the w alternate routes are examined.

If the alternate route improves the network cost, it is newly adopted and go to Step 3-2.

Step 4. If $l < |L|$, then set $l = l + 1$ and go to Step 3. Else go to Step 5.

Step 5. If the network cost is improved in Step 3 and 4, then go to Step 1. Else stop.

E. Comparison of the Proposed and Alternate Algorithm

Given backup VP routes for all demands, the Alternate algorithm calculates total network cost by computing the required spare capacity $S(l)$ on each link l. Note that the same process is used to compute total network cost at Step 5 of each proposed algorithm.

An initial and two improvement heuristics are proposed based on properties in Section III-B. Each heuristic first checks if two target VPs are link disjoint or not. If those target VPs are link disjoint in the network, the network cost is reduced by routing those corresponding two backup VPs through a common path. This is the difference which is not employed in the Alternate algorithm. Also the cost improvement process is performed at Step 3-2 by adopting the "modified link cost" for the links in the common route in each proposed algorithm.

As an illustration, consider two target demands k and $k+1$ in Fig. 2(a). In the figure, working paths WP(k) and WP(k+1) are link disjoint. If the Alternate algorithm is applied, two backup paths BP(k) and BP(k+1) may have different routes by the shortest path method. In that case, if two backup paths have no common links, each link on BP(k) requires bandwidth b^k and each on BP(k+1) requires b^{k+1}. However, in the proposed algorithms, since two working paths are link disjoint, the proposed algorithm provides a common backup paths by routing BP(k) and BP(k+1) as shown in the figure. Thus no additional bandwidth is required for the links in the common route.

IV. Computational Results

To illustrate the efficiency of the algorithms proposed in this paper, three example networks are used which are presented in [13]. Topological information for those networks are given in Table I. Example networks are shown in Figure 3-5. Note that in the network a circle and a square together represents an ATM switching node which is co-located (same city) with an ATM cross-connect node. The link between two cross-connect nodes is the possible link placement for which spare capacity needs to be determined. The distance of the link between two nodes is ranged from 10 to 500 miles according to the network topology.

Each demand pair has a working VP and requires one or two backup VPs. Note that the working VP

is required between any two ATM switch nodes. For each demand pair, the required bandwidth is generated using a uniform random number generator ranged from 1 to 50 Mbps. Thus five instances are generated with different demand requirements for each problem. In fact, the required bandwidth or equivalent bandwidth is estimated more accurately in the dimensioning problem by considering many traffic descriptors and QoS requirements [2]. We have used the modular grouping value, a, for capacity unit of a link to be 1.5 Mbps which is equivalent to T1 rate (Note that this link speed is used for only 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 the port cost at the nodes, we have computed the unit cost of links of 1.5 Mbps using 5 as the cost of each termination port and 0.02 to be the distance cost per mile. Thus, $c_{ij} = 2 * 5 + 0.02 * D_{ij}$, where D_{ij} is the distance in miles for 1.5 Mbps for link (i,j).

All the computational results in this section are coded by C language and tested on WorkStation HP 9000/715 Series. Our computational work is aimed at addressing the following items: 1) the comparison of performance for the algorithms proposed in this paper, 2) the convergence rate of the proposed algorithm to an optimal solution or the best available lower bound 3) the gap of the solutions obtained by the algorithms from the best available lower bound, 4) the comparison between the network cost and the worst-case survivability in multiple link failures as the number of required backup VPs increases.

Table II compares the performance of two initial algorithms: the shortest path algorithm and WVPBA for the problems with single and double backup VPs. From the tables, it is clear that the WVPBA heuristic provides a better performance than the shortest path heuristic [6] as an initial algorithm for all problems. Thus the WVPBA is employed as an initial solution procedure for all experiments to follow.

Table III represents the performance of three improvement heuristics: Alternate [6], BVPBA and MLBA for the problems with single and double backup VPs. In BVPBA and MLBA heuristics, $w = 5$ is used for the window size. In other words, five alternate routes are examined to improve the network cost in the algorithms. The value in the parenthesis represents the CPU second. The Alternate and the BVPBA heuristics show almost same performance. However, it is clear that the MLBA dominates other improvement heuristics in all problems. The MBA in the table represents the result obtained by applying three improvement algorithms. For the implementation of the MBA heuristic, we employ two different orders. First, the MBA is implemented in the order of Alternate, BVPBA and MLBA which is the ascending order of improvement range. The second procedure employed is descending order: MLBA, BVPBA and Alternate. In most of our experiments, it is shown that the solution quality of the descending order outperforms that of the ascending order. Thus the descending order is employed to have the computational results of the MBA procedure. From Table III, it is clear that the MBA improvement heuristic with an initial solution obtained by the WVPBA heuristic provides the best performance.

To examine the performance of MBA, the best available lower bounds are obtained. CPLEX Integer Programming Solver [22] is used to solve the integer program in Section II. However, we could not

obtain the optimal solutions even for problems with 10 nodes within three hours. Thus we obtained the best available lower bound by running the CPLEX for three hours. Table IV and V show the result of Alternate (Alternate heuristic with an initial solution by Shortest) [6], MBA and the best available lower bound for each instance with single and double backup VPs, respectively. From the tables, we see that the MBA produces better solutions than Alternate for all instances. In addition, compared to Alternate algorithm, the MBA provides a significant performance improvement as the number of nodes and backup VPs increase.

Figure 6 shows the convergence rate of the MBA algorithm for the problems with single backup VP. In the figure, the number of iterations represents total number of running times of individual heuristics, i.e., Step 1- 4 of the BVPBA and MLBA heuristics. From the figure it is clear that as the number of nodes increases, the convergence rate of the algorithm becomes slower.

Figure 7 shows the gap of the solution by the MBA heuristic from the best available lower bound as the number of required backup VPs increases. To generate problems with different number of required backup VPs, all demands are sorted in nonincreasing order of the bandwidths requirement for each VP. Each VP is assumed to require at least one backup and high-priority VPs require additional one backup VP. As an example, the value 1.5 in the x-axis of Figure 7 implies that one half of the total VPs require two backup VPs. In the figure, the gap from the best available lower bound (LB) in the y-axis represents a relative cost of $100(MBA-LB)/LB$. As the number of nodes and the required backup VPs increase, the gap becomes larger from the best available lower bound. On the average, the gap from the best available lower bound is ranged from 2% to 11%.

Figure 8 represents the ratio of the network cost and the survivability for the problems with 10 nodes with various number of backup VPs. In the figure, the survivability represents the worst-case ratio of the restored bandwidth by the backup VP to the total required bandwidth, when two simultaneous link failures occur in the network. Note that 100% survivability is achieved when two backup VPs are employed for all demands. The ratio of the network cost represents the relative cost of the network with double backup VPs, compared to that with single backup VP. From the figure we see that both the network cost and the survivability increase as the number of required backup VPs increases. However, it is clear that the survivability is improved significantly, compared to the increase of the network cost. As an example, 100% survivability is achieved in the network with two backup VPs with 70% additional cost required for the network with single backup VP.

V. Conclusions

This paper proposes a design of self-healing ATM networks based on the backup virtual path. Under normal operation of an ATM network no bandwidth is assigned to backup VPs. However, when a link fails, the backup VPs become active to restore the routes of target demands affected by the failed link.

13

Given the bandwidth requirement and the working route for each demand, the problem is to find the backup routes of each demand that is link disjoint from the working route. The required spare bandwidth for the backup routes also need to be determined at each physical link such that the cost of total spare bandwidth is minimized. This problem is formulated as an integer linear programming model.

As an initial heuristic, the working VP based algorithm is proposed which employs the structural properties of the backup routes of any two distinct demands. The backup route of each demand is determined such that it shares the route of other backup whenever it exists. Two improvement algorithms are also developed. The backup VP based algorithm improves the initial solution by finding alternate backup VP with reduced cost. The maximum link based algorithm, on the other hand, improves the network cost by rerouting the backup VPs which pass through the link with the maximum link cost.

Computational experiments show that the gap between the proposed solution and the optimal solution is less than 11%. For all problems tested the proposed algorithm improves the initial solution with a gap of 2 - 11% from the best available lower bound. The computational experiments also show that the worst case survivability can be improved significantly with relatively small increase of bandwidth.

References

[1] E. Ayanoglu and R. D. Gitlin, "Broadband Network Restoration", IEEE Communication Magazine, July, 1996.

[2] A. I. Elwalid and D. M. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks", IEEE/ACM Trans. on Networking, Vol. 1, No. 3, 1993.

[3] S. P. Evans, "Optimal bandwidth management and capacity provision in a broadband network using virtual paths", Performance Evaluation, Vol. 13, pp. 27 - 43, 1991.

[4] S. Hasegawa, Y. Okanoue, T. Egawa and H. Sakauchi, "Control algorithm of SONET integrated self-healing networks", IEEE JSAC, Vol. 12, No. 1, pp. 110 - 119, 1994.

[5] R. Kawamura, H. Hadama and I. Tokizawa, "Implementation of Self-healing Function in ATM networks based on virtual path Concept", Proceeding of IEEE INFOCOM '95, Boston, Apr. 1995.

[6] R. Kawamura, K. Sato and I. Tokizawa, "Self-healing ATM networks based on virtual path concept", IEEE JSAC, Vol. 12, No. 1, pp. 120 - 127, 1994.

[7] R. Kawamura and I. Tokizawa, "Self-healing virtual path architecture in ATM networks", IEEE Communication Magazine, September, 1995.

[8] S. B. Kim, M. J. Kim and S. I. Lee, "Mathematical models for dimensioning of ATM networks", IEEE Globecom, pp. 240 - 244, 1995.

[9] S. J. Koh and C. Y. Lee, "A tabu search for the survivable fiber optic communication networks design", Computers and Industrial Engineering, Vol. 28, No. 4, pp. 689 - 700, 1995.

[10] T. V. Landegem, P. Vankwikelberge and H. Vanderstraeten, "A self-healing ATM network based on multilink principles", IEEE JSAC, Vol. 12, No. 1, pp. 139 - 148, 1994.

[11] C. Y. Lee and S. G. Chang, "Balancing loads on SONET rings with integer demand splitting", Computers and Operational Research, Vol. 24. No. 3, pp. 221-229, 1997.

[12] C. Y. Lee and S. J. Koh, "A design of the minimum cost ring-chain network with dual-homing survivability: a tabu search approach", Computers and Operational Research, Vol. 24, No. 9, pp. 883-897, 1997.

[13] D. Medhi, "Multi-hour, multi-traffic class network design for virtual path-based dynamically reconfigurable wide-area ATM networks", IEEE/ACM Trans. on Networking, Vol. 3, No. 6, 1995.

[14] D. Medhi, "A unified approach to network survivability for teletraffic networks: model, algorithms and analysis", IEEE Trans. on Communication, Vol. 42, pp. 534 - 548, 1994.

[15] K. Murakami and H. S. Kim, "Virtual Path Routing for Survivable ATM networks", IEEE/ACM Transaction on Networking, Vol. 4, No. 1, pp. 22 - 39, 1996.

[16] R. O. Onvural, "Asynchronous transfer mode networks: performance issues", 2nd edition, Artech House, 1995.

[17] Y. Sato and K. I. Sato, "Virtual path and link capacity design for ATM networks", IEEE JSAC, Vol. 9, pp. 104 - 111, 1991.

[18] T-H. Wu, "Fiber network service survivability", Artech House, 1992.

[19] J. Yan and M. Beshai, "Designing an ATM-based broadband network: an overview", IEEE lobecom, pp. 245 - 252, 1995.

[20] ITU-T Recommendation I.150, "B-ISDN asyncronous transfer mode functional Characteristics", 1992.

[21] ITU-T Draft Recommendation I.610 (revised), "B-ISDN operation and maintenance principles and functions", 1994.

[22] CPLEX version 2.1, CPLEX Optimization, Inc, 1993.