A Fault-tolerant Multicast Routing Algorithm in ATM Networks

S. Kwong
Computer Science Dept.
City University of Hong Kong
83 Tatchee Avenue, Hong Kong

S. S. Chan
Computer Science Dept.
City University of Hong Kong
83 Tatchee Avenue, Hong Kong

Abstract

This paper presents a genetic algorithm based method to solve the capacity and routing assignment problem arising in the design of self-healing networks using the Virtual Path (VP) concept. Past research has revealed that Pre-planned Backup Protection method and the Path Restoration scheme can provide a best compromise on the reserved spare capacity and the failure restoration time. Base on a set of customer traffic demands, we will determine the routings of the working and backup virtual paths to satisfy the demands, so that the traffic is 100% restorable under single point of failure and at the same time the amount of spare capacity can be minimized. In contrast to the past researches on this area, we will use Genetic Algorithm instead of linear programming. There are several advantages of using Genetic Algorithm which included faster and lower computation cost in reaching a reasonably good virtual path routing scheme, it can tackle multiple objectives function effectively, and less complex mathematical formulation. We will show that the results obtained using Genetic Algorithm is better than those results obtained by using heuristic approaches. Another contribution of this paper is our method not only can work on the Unicast traffic context just as the past researches, but also Multicast traffic.

1 INTRODUCTION

We are entering to an Information Age in that information is the king. In view of such demands, companies have kept investing on the infrastructure of different information technology systems. It is for sure that a standalone closed information system cannot help the company to survive in these times. The intensive cooperation of different information systems fuels a massive networking demand. To cope with the increasing networking demands from the customers, currently TELCO operators are changing their networks from coaxial cables to optical fibers, transmission technology from PDH (plesiochronous digital hierarchy) to SDH (synchronous digital hierarchy), and switching technology from STM (synchronous transfer mode) to ATM (asynchronous transfer mode).

The deployment of fiber optics in ATM / SDH network leads to the fact that more and more traffic is concentrated on a few of fibers, why? For most companies, the bandwidth of a fiber is far bigger than their aggregated traffic demands. So to use fiber in a cost-effective manner, it is usual for the TELCO to multiplex a lot of different customer traffics together onto a fiber. Fiber is expensive and a fiber already can satisfy the aggregated traffic demands of a customer, so all the traffic data of the customer will be carried over a single fiber. It implies the data communication network of the customer is highly dependent on the healthiness of the fiber. On the TELCO side, since one fiber cable can replace several tens coaxial cables, the number of cables in the trunk and international backbone networks of the TELCO will be reduced a lot.

The implementation of high capacity fiber based ATM / SDH network, although providing a lot of advantages through multiplexing and sharing of transport facilities, it increases the vulnerability of telecommunication services using these transport facilities. Communication networks are potentially under attacks by many things anytime and anywhere: varying from natural disasters and accidents to erroneous hardware and software. If there is a breakdown in the fiber network (e.g. a fiber cut or a fiber node equipment malfunction), a lot of customers will be affected. Both the customers and the TELCO may then suffer from a huge loss of revenue. Therefore, the capability of the fiber network to restore services affected by failures within the shortest possible time thus becomes a key network architecture design consideration.

To restore affected traffic upon a network failure within a short time duration, a usual practice is to pre-allocate redundant (spare) capacity in the network. By doing so, when the network management system (NMS) detects any kind of failure in the ATM network, traffics can be re-routed (via the spare capacity) around the failed spot automatically and gracefully. If sufficient spare capacity is reserved in the ATM network, all the traffics can be re-routed automatically upon failure and the customer applications will not notice there is a network problem, and the network is said to be a "*self-healing*" network. The term self-healing refers to the capability of the network to reconfigure itself around failures quickly and gracefully with the goal of approaching 100% service survivability on an end to end basis. Survivability refers to the ability of the network to provide continuity of service with no disruption. However, equipping redundancy or spare capacity against fiber network failures could be very expensive due to the high cost associated with the fiber and the fiber transmission equipment. If there is too much spare capacity, the TELCO operator may not run the network cost-effectively. On the other hand, the protection level may not be enough. So, what should be the optimal amount of spare capacity needs to deploy? Reducing network protection costs while maintaining an acceptable level of survivability is the main objective of the network planners and engineers. To save cost, nodes within the network are not fully connected with each other. Links between nodes are not all of same speed. The transmission delay from a source node to a destination node depends on the paths chosen – routing. Every customer likes to have a route with minimum delay but for cost issue, TELCO operator cannot promise to every customer that his route is the shortest. So, how to choose routes which will compromise the interests of the TELCO operator and different customers?

To tackle the above two problems, a method based on Genetic Algorithm is presented in this paper. We will see how the method can be used to find out the optimal level of working and spare capacity of a self-healing ATM / SDH network in order to provide a highly available services. We will see during the determination of optimal working and spare capacity, the quality of routing will not be neglected.

2 RELEATED WORK

The capacity and routing assignment arising in the design of self-healing network is an optimization problem, i.e., optimizing capacity allocation and routing assignment to minimize the cost.

In [Chen 1996], an efficient spare capacity allocation strategy is proposed to optimise the amount of and the location of spare capacity in the network based on link level such that it meets the self-healing criteria. The basic principle of this method is that the spare capacity of every link is shared for survivability purpose. This method takes the spare capacity of each link as the object of manipulation and three constraint equations are derived. The three constraint formulas are used for the establishment of a new Working Virtual Path (WVP) or a new Backup Virtual Path (BVP), with respect to two parameters: the spare capacity ratio[1] and the spare capacity share ratio[2] of each link. This method employs the Pre-planned Backup Virtual Path Protection scheme [Xiong 1999] and has no guarantee on the quality of the WVP and BVP. In addition, it only deals with the context of unicast traffic, but not on multicast traffic.

[Murakami 1998] proposed to use the linear programming (LP) technique to solve this optimization problem. The basic idea of this scheme is find an optimal capacity placement under a given a projected traffic demand such that capacity and flow assignment is jointly optimized. The problem is formulated as a large-scale linear programming. The basis matrix is factorized into a LU form by taking advantage of its special structure, which results in a reduction of computation time. The problem with this method is the highly intensive computation is required. Moreover, the aim of this method is to minimize the cost, and it may produce routes of a WVP or BVP that is unacceptably long and fails to meet the QoS of the customer. In addition, the focus of the method is on unicast traffic only.

[Iraschko 1998] further proposed to use an integer programming technique to determine the optimal spare capacity placement for self-healing networks. It also employs the linear programming (i.e. integer programming) over a set of constraints and to optimize the placement of only the spare capacity when the working demands are already routed, or jointly optimize both working and spare capacity, of either a link- or path-restorable network. It uses flow constraints based on a set of eligible predefined routes over which path sets may be implemented. This method also suffers from the usual disadvantages of linear / integer programming: a lot of constraints need to setup and a lot of variables need to manipulate. It needs intensive computation. Moreover, it concentrates on unicast traffic and just optimizes the capacity placement only: it cannot optimize the traffic routings at the same time.

[Xiong 1999] formulated the same problem as a linear programming problem. This is a very good and comprehensive paper on the capacity placement and routing assignment problem in self-healing network. It has compared several restoration strategies quantitatively in terms of spare cost: global versus failure-oriented reconfiguration, path versus link restoration, and state dependent versus state independent.

As can be seen, most of researches mentioned above is to view the capacity and routing assignment arising in the design of self-healing network as an optimization problem, i.e., optimizing capacity allocation and routing assignment to minimize the cost and solve it using linear programming techniques. They also only aimed to solve the unicast problem only. Designs for telecommunication networks that can minimize cost while satisfying traffic requirement for unicast traffic is NP-hard [Dengiz 1997]. It is also well known that it has a number of local optima that are far away from the global optimal, in this case, these linear programming, integer programming or mix integer programming techniques are found to be ineffective [Davis 1993]. On the other hand, the use of genetic algorithm in designing mesh network has proven successfully for the unicast traffic. In [Ko 1997], the

[1] The spare capacity ratio of a link is the ratio of the total spare capacity on the link to the total capacity of WVP's on the link.

[2] The spare capacity share ratio of a link is the ratio of the total virtual capacity of BVP's on the link to the total spare capacity on the link.

authors presented a total solution to mesh network design. It does not optimize network topology, but also the routing and capacity assignment. Dengiz etal [Dengiz 1997] also proposed to use genetic algorithm to solve the NP-hard problem for reliable network design. Davis et al also proposed to use genetic algorithm to solve the survivable network problem and it has a better performance when comparing with some greedy algorithms [Davis 93]. However, their problem is not for the multicast traffic and only deal with bandwidth allocation. In [Zhang 1999], the authors proposed an orthogonal genetic algorithm for multicast routing assignment problem. It proposes an Orthogonal Genetic Algorithm to determine a minimum cost multicast path from a single source node to multiple destination nodes. It targets on a single multicast path, NOT on many multicast paths to satisfy all the customer demands. The paper does not do any optimization on the overall working and spare capacity placements.

In fact, most of the works done in the past based on the followings:

1. Pre-planned Backup Virtual Path Protection self-healing scheme was used.
2. Linear or integer programming was used as a mean to solve the optimization problem.
3. Their target of interest is unicast traffic.
4. Their objective is to achieve minimum working and spare capacity.

In general, they also have some common weaknesses:

1. They cannot handle multicast traffic and its backup.
2. They cannot optimize the capacity and the quality (e.g. delay) of the routing paths at the same time.
3. Intensive computation is needed (time in terms of days may need to figure out the optimal capacity level).
4. It is almost impracticable for a relatively large network
5. Either the optimal solution or none is provided for the user. It cannot let the user to sacrifice the optimal degree for reduced computation time.
6. It may get trapped in local extreme – not global extreme
7. The objective function and the constraints need to be linear functions of the designed variables.

Our objective of this paper is to study a method that can overcome or reduce the above weaknesses:

1. We will employ Genetic Algorithm instead of linear or integer programming as a mean to do the optimization of the capacity and routing assignment problem. By employing Genetic Algorithm, we can avoid most of the problems inherent in linear programming.
2. A fitness function incorporates factors of the total working and spare capacity of the network and the quality of the routings will be used. We can optimize capacity and quality of the routings at the same timeWe will model a sub-gene, a gene and a chromosome such that the GA system can handle any multicast traffic demands.
3. We will also design means to model a self-healing network and the customer traffic demands. Users are required to input the model characteristics of the network and traffic into the GA system. Based on the input network and traffic, the GA system will employ different genetic operations and output an optimal solution plus a set of non-optimal solutions to the users within the users' specified time period (i.e. in terms of generations).

3 PROBLEM FORMULATION

The following assumptions are made for this research :

- The target network of study is an ATM network.
- Single point of failure in the network is assumed.
- It is assumed the network can restore all the affected traffics under single point of failure.
- Each branch of a multicast circuit requires the same bandwidth (however, different multicast circuits can have different bandwidths)
- Link transmission delay is directly proportional to the link distance

The following notations are used in the problem formulation:

$\mathcal{N}$ Set of nodes of the network

$\mathcal{A}$ Set of directed arcs of the network

C_a Capacity on arc a, where $a \in \mathcal{A}$

Π Set of source-destinations multicast traffic requirements

$\mathcal{WVP}$ Set of multicast working virtual paths

$\mathcal{BVP}$ Set of multicast backup virtual paths

$|\Pi|$ Total number of multicast traffics demanded

$|\mathcal{WVP}|$ Total number of multicast working virtual paths

$|\mathcal{BVP}|$ Total number of multicast backup virtual paths

Π_i i-th source-destinations multicast traffic in Π

WVP_i i-th multicast working virtual path in $\mathcal{WVP}$ to satisfy Π_i

BVP_i i-th multicast backup virtual path in $\mathcal{BVP}$ to backup WVP_i

$WVP_i(a)$Capacity of arc a would be used by WVP_i, where $a \in \mathcal{A}$

$BVP_i(a)$ Capacity of arc a would be reserved by BVP_i, where $a \in \mathcal{A}$

WVP_i.delay Total transmission delay of WVP_i

BVP_i.delay Total transmission delay of BVP_i

$O(\Pi_{ij})$ Source node of the j-th branch of Π_i

$O(WVP_{ij})$ Source node of the j-th branch of WVP_i

$O(BVP_{ij})$ Source node of the j-th branch BVP_i

$D(\Pi_{ij})$ Destination node of the j-th branch of Π_I

$D(WVP_{ij})$ Destination node of the j-th branch of WVP_i

$D(BVP_{ij})$ Destination node of the j-th branch BVP_i

Π_{ij}.bandwidth Bandwidth demanded by j-th branch of Π_i

WVP_{ij}.bandwidth Bandwidth of j-th branch of WVP_i

BVP_{ij}.bandwidth Bandwidth of j-th branch of WVP_i

I-WVP_{ij} It is the intermediate virtual path of WVP_{ij} – with source and destination nodes excluded.

I-BVP_{ij} It is the intermediate virtual path of BVP_{ij} – with source and destination nodes excluded.

The multicast routing problem can be described as follows.

Objective: To find $\mathcal{WVP}$ to satisfy the requirement Π, and the $\mathcal{BVP}$ to provide alternative routes which can restore all the failed multicast working virtual paths in $\mathcal{WVP}$ under single point of failure of the network, and gives:

$$Min\left\{k \sum_{a \in A} \sum_{i=1}^{|\Pi|} \left[WVP_i(a) + BVP_i(a)\right] + (1-k) \sum_{i=1}^{|\Pi|} \left[WVP_i.delay + BVP_i.delay\right]\right\}$$

where k is an integer from 0 to 1. It depends on how you like to compromise the two factors: capacity and delay.

Subject to constraints:

1. $|\Pi| = |\mathcal{WVP}| = |\mathcal{BVP}|$ AND

$O(\Pi_{ij}) = O(WVP_{ij}) = O(BVP_{ij})$ AND

Π_{ij}.bandwidth = WVP_{ij}.bandwidth = BVP_{ij}.bandwidth

where $\forall$ i = 1 to $|\Pi|$ and j = 1 to degree of multicast of i-th multicast traffic demand

2. I-$WVP_{ij} \cap$ I-$BVP_{ij} = \varnothing$
in link- or node- disjointed sense.

where $\forall$ i = 1 to $|\Pi|$ and j = 1 to degree of multicast of i-th multicast traffic demand

3.

$$\sum_{i=1}^{|\Pi|} \left[WVP_i(a) + BVP_i(a)\right] \le C_a \qquad \forall a \in A$$

Constraint (1) ensures that the $\mathcal{WVP}$ found can satisfy Π, and the $\mathcal{BVP}$ can support $\mathcal{WVP}$ under a single point of failure of the network. Constraint (2) is to make sure backup virtual path is completely node / link disjointed from the working virtual path. Constraint (3) is to make sure the capacity of any arc can cover the bandwidth used by $\mathcal{WVP}$ and reserved by $\mathcal{BVP}$.

4 DESIGN OF GENETIC ALGORITHM

In this research, we will model the network as an undirected graph $G = (V, E)$, where V is a set of nodes and E is a set of links. Each link is a pair of oppositely directed arcs or fibers. We don't model the network as a directed graph just because most of the communication networks are generally bi-directional. Each site in the network will be represented as a node. Each node will be assigned a number as its ID. From now on, if not specified, we assume the bandwidth of each link is ***B*** unit.

4.1 CHROMOSOME ENCODING

In genetic algorithm, a chromosome is a potential solution to the problem. In this problem, the solution is a set of working virtual paths (WVP's) and a set of backup virtual paths (BVP's) for the multicast traffic, which can minimise the overall working and spare capacity and the overall transmission delay of the network. If a gene is made to be a set of WVP's and BVP's which satisfy a particular customer multicast traffic demand, then we can model a chromosome as a collection of genes (Fig. 1).

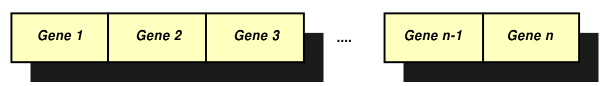


Fig. 1: Model of a Chromosome

Please be noted that the number of genes in the chromosomes should be equalled to the total number of customer traffic demands.

A gene is a solution to a particular multicast traffic demand. We can view a multicast traffic as a number of unicast traffics leaving from the same source node but different destination nodes. If we design an entity (let's call it sub-gene) to represent a solution for a unicast traffic, then we can model a gene as a collection of the entities. In the proposed GA solution, we model the virtual path (no matter it is WVP or BVP) as a string of the following format:

sourceNode-node2-node3-…-nodeN-destinationNode

The number of sub-genes in a gene is equal to the degree of multicast of a particular multicast traffic demand. If you think carefully, a sub-gene consists of a WVP and a BVP only to satisfy a particular unicast traffic demand.

4.2 POPULATION POOL INITIALIZATION

Given the topological information of an ATM network, plus a source node and a destination node. How can we find out all possible paths from the source node to the destination nodes? Notes that each possible path should not include any intermediate node more than one. This condition is to make sure all the paths generated will not waste unnecessary resources. We implemented following prototype *VP_explore* for this purpose.

void VP_explore(FILE* fp, int source, int destination, char* u_path);

where fp= Pointer to a file which will contain all the possible paths from the source node to destination node

source = source node

destination = destination node

u_path = an intermediate path from the source node

The details of theimplementation is as follows:

1. Set ***source*** = sourceNode, ***destination*** = destinationNode, ***u_path*** = "sourceNode-"

   Open the file ***souce-destination.VP*** for writing and set ***fp*** to be a file pointer to the file.

   Activate the procedure *VP_explore* (***fp. source, destination, u_path***).
2. Find all the connected neighbour nodes (i.e. nodes that with physical links from the node ***source*** to them) of the ***source***.
3. If all of the neighbour node(s) of the ***source*** have been "selected", return; else pick one neighbour node, say ***S***, mark it as "selected".
4. If ***S*** == ***destination***, set ***u_path*** := ***u_path*** + ***destination***, and saves ***u_path*** to the file ***fp***, and go to (2).

   Else if ***S*** is already included in the ***u_path***, ignore ***S***, go to (2) again.

   Else set ***source*** = ***S***, and activate *VP_explore* (***fp. source, destination, u_path***). After return from *VP_explore*, go to (2)

After getting all the possible VP's for a branch of a multicast circuit, we will distribute various VP's to the sub-gene of various chromosomes of the initial population to increase the *"diversity"* of the chromosomes. Notes that a pair of link or node disjointed VP's (i.e. a working VP and a backup VP) needs to allocate to each chromosome.

When all the sub-genes of a chromosome have been allocated pairs of VP's, we then need to ensure that the chromosome generated is a valid one. A valid chromosome should posses the following properties:

- Depends on which path disjoint condition we are using, each pair of working and backup VP's should be node-disjointed or link-disjointed from each other. To avoid a single point of failure of either a link or a node of an ATM network from causing both the working VP and its corresponding backup VP to be failed at the same time, the pair of working and backup VP's needs to be link-disjointed or node-disjointed.
- For each physical link used by the VP's of the chromosome, the capacity of the link should be able to cater for all the working VP's traffic demand and the bandwidth to be reserved by the backup VP's

4.3 FITNESS COMPUTATION

After all the chromosomes of a generation are created, the fitness of each chromosome needs to be computed for later genetic operation purpose. The fitness function for the GA system is defined as follows:

$$Fitness = \frac{K\_M}{\left\{\sum_{j=1}^{L}(Ka\_W * C_j + Ka\_S * S_j)\right\} + \left\{\sum_{i=1}^{N}(Kb\_W * D_i + Kb\_S * T_i)\right\}}$$

Where

L	is the total number of links in the network
C_j	is the capacity needed to reserve in link j for the WVP's
S_j	is the capacity needed to reserve in link j for the BVP's
N	is the total number of multicast circuits
D_i	is the transmission delay of i-th working multicast circuit
T_i	is the transmission delay of i-th backup multicast circuit
K_M	is a constant multiplier for scaling up and down the fitness
Ka_W	is a weight factorfor C_j
Ka_S	is a weight factor for S_j
Kb_W	is a weight factor for D_i
Kb_S	is a weight factor for T_j

Notes that in general *K_M* will be 1, *Ka_W* = *Ka_S*, *Kb_W* = *Kb_S*, and *Ka_W* +*Kb_W* =*1* or *Ka_S* +*Kb_S* =*1*.

For the capacity optimization problem, the values of *Ka_W and Ka_S* in the above fitness function are set to 1 (*K_M = 1 and Kb_W* = *Kb_S* = *0*). On the other hand, if we are going to study the shortest path (i.e. delay) optimization problem, then the value of *Ka_W and Ka_S* is set to 0. If we are studying a problem to compromise the capacity and delay at the same time, *Ka_W*, *Ka_S*, *Kb_W*, *Kb_S* are set to values between 0 and 1.

The fitness computations are responsible for computing the working and backup capacity usage; and the_delays of the working and backup VP's. Since for a multicast circuit, it is highly probable that the delays of all its branches will not be same. We will select the longest delay of the branches to be the delay of the multicast circuit, i.e.,

D_i = *Longest* transmission delay of the branches of the i-th working multicast circuit

T_i = *Longest* transmission delay of the branches of the i-th backup multicast circuit

4.4 CROSSOVER

We have designed three types of crossover operations, namely:

1. Chromosome Crossover
2. Direct Gene Crossover
3. Indirect Gene Crossover

A crossover is said to be valid if at least one of the above three crossover operations give two valid offspring chromosomes. Whereas, a crossover is invalid if none of the above three crossover operations can give a pair of valid offspring chromosomes. The strategy of using the above three crossover operations is to use one type of crossover operation first, if the crossover is valid, then the crossover mechanism of the chromosomes pair is said to be completed; otherwise, we will try the other types of crossover operations.

There are six combinations to arrange the above three crossover operations, namely:

1. Chromosome, Direct Gene, Indirect Gene Crossovers
2. Chromosome, Indirect Gene, Direct Gene Crossovers
3. Direct Gene, Chromosome, Indirect Gene Crossovers
4. Direct Gene, Indirect Gene, Chromosome Crossovers
5. Indirect Gene, Chromosome, Direct Gene Crossovers
6. Indirect Gene, Direct Gene, Chromosome Crossovers

The above lines listed the ascending order of types of crossover operations that will be performed. To avoid using one type of crossover operation more than the rest, each time when a crossover is to be performed, one of the six crossover orders will be selected randomly.

4.4.1 Chromosome Crossover

The purpose of the *Chromosome Crossover* is to explore whether the different combinations of genes will result in better or fitter chromosomes. This type of crossover just involves direct exchange of genes between two selected chromosomes. Since a chromosome has ***n*** genes, there are (***n-1***) potential crossover points in the Chromosome Crossover. To be fair, the crossover point ***Gene m*** of the two parents chromosomes is selected randomly. Then the genes left to ***Gene m*** of Parent 1 will combine with ***Gene m*** (of Parent 2) and genes right to the ***Gene m*** in Parent 2 to form Offspring 1. Similarly, the genes left to ***Gene m*** of Parent 2 will combine with ***Gene m*** (of Parent 1) and genes right to the ***Gene m*** in Parent 1 to form Offspring 2. If the offspring chromosomes are all valid, then the parent 1 and parent 2 chromosomes in the mating pool will be replaced by them and the crossover operation of the two parents will be completed. Otherwise, another crossover point will be considered. If all the crossover points have been tried, but valid offspring chromosome still cannot be produced, then the Chromosome Crossover is declared to be INVALID. If the Chromosome Crossover is INVALID, then the other types of crossover operations will be performed. If all of the types of crossover operations are INVALID, then the crossover of the two parent chromosomes is said to be INVALID.

4.4.2 Direct Gene Crossover

For Direct Gene Crossover, the initial available number of crossover points is equalled to the multicast degree minus one. This type of crossover is aimed to create new possible genes – and thus new chromosomes. You may notice that the above operation cannot be carried out when the degree of multicast of a gene is 1 (i.e. unicast): the number of available crossover point is zero at the beginning. In that case we will directly exchange the WVP and BVP between the two genes of the two parents. Suppose,

Parent 1: i-th gene <a_WVP_1, a_BVP_1>

Parent 2: i-th gene <***A_WVP_1, A_BVP_1***>

Then,

Offspring 1: i-th gene <a_WVP_1, ***A_BVP_1***>

Offspring 2: i-th gene <***A_WVP_1***, a_BVP_1>

We will say the Direct Gene Crossover operation is VALID if two valid offspring chromosomes are produced. In that case, the two offspring chromosomes will replace the two parent chromosomes in the mating pool and the crossover operation is completed. If two valid offspring chromosomes cannot be produced, other types of crossover (Chromosome Crossover or Indirect Gene Crossover) may be performed. If all of the types of crossover operations fail to produce valid chromosomes, then the crossover of the two parent chromosomes is said to be INVALID.

4.4.3 Indirect Gene Crossover

This type of crossover mechanism is not brought from the conventional Genetic Algorithm. It is dedicated to the capacity and routing assignment problem of a self-healing network. We name it *Indirect Gene Crossover* just because there is not whole gene / sub-gene directly exchanged between two parent chromosomes. Instead, part of the sub-gene will be exchanged between two parent chromosomes. The details of the algorithm is described as follows:

1. Let p1, p2 be the two selected parent chromosomes; c1, c2 be the two offspring chromosomes; N = number of genes; D(I) = degree of multicast of I-th gene; p1(I,J) = J-th subgene of I-th gene of parent1; similarly for p2(I,J); c1(I,J) = J-th subgene of I-th gene of offspring1; similarly for c2(I,J); VALID_CROSSOVER=FALSE; I = 1.
2. If I > N, then return VALID_CROSSOVER and stop; Else J = 1.
3. If J > D(I), I = I + 1, then goes to (2).

4. Search the WVP's of p1(I,J) and p2(I,J) subgenes for any common intermediate nodes (other than the source and destination nodes).

5. If common intermediate node(s) exists, exchange the sub-paths of WVP's (using the common intermediate node as the crossover point) of the p1(I,J) and p2(I,J) to form two new WVP's (i.e. c1(I,J) and c2(I,J)). If c1 and c2 are valid, set VALID_CROSSOVER=TRUE, J = J + 1, then goes to (3).

6. Search the BVP's of p1(I,J) and p2(I,J) subgenes for any common intermediate nodes (other than the source and destination nodes).

7. If common intermediate node(s) exists, exchange the sub-paths of BVP's (using the common intermediate node as the crossover point) of the p1(I,J) and p2(I,J) to form two new BVP's (i.e. c1(I,J) and c2(I,J)). If c1 and c2 are valid, set VALID_CROSSOVER=TRUE, J = J + 1, then goes to (3).

8. J = J + 1, goes to (3) again.

4.5 MUTATION

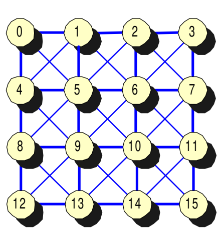
Mutation is another important activity in genetic algorithm. It aims to introduce some changes to the existing chromosomes to avoid them to be trapped in a local optimum. The mutation operation will be performed only after crossover operation is complete. We will not apply mutation operation to all the chromosomes in the population. Instead, we will base on the mutation probability (i.e. MUTATION_PROBABILITY) to select a set of chromosomes for mutation operation. A gene is said to be mutated or changed if at least one of its WVP's or BVP's has been replaced by a new one. When a gene is changed, the chromosome is also changed. If the resulting chromosome is a valid one, then it will replace the selected chromosome; else no change to the selected chromosome.

5 EXPERIMENTAL RESULTS

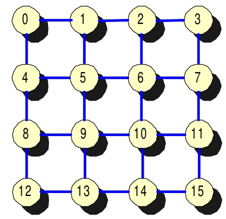
We have carried out a few experiments to demonstrate the effectiveness of our algorithm for self-healing network design problem. This Section will describe how we carry out the performance study and the results.

The BPR (Backup Capacity Usage to Primary or Working Capacity Usage Ratio) would be computed and used as a measurement of merit. The lower the BPR value, the better the result. We have used the following four networks for the study (Fig. 2).

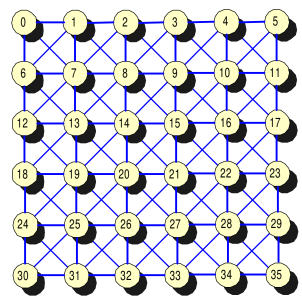
Network 1



Network 2



Network 3



Network 4

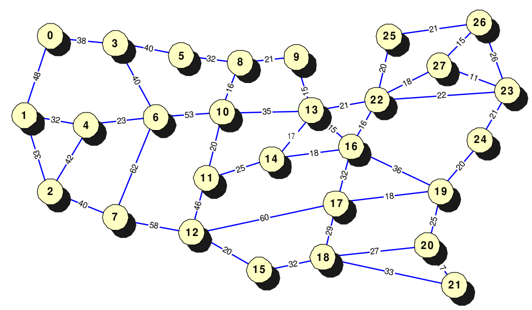


Fig. 2: <u>Networks for Study</u>

We also have implemented the link restoration and path restoration schemes using heuristic approach for comparison purposes. The results are plotted in Figs. 3a to 3d. It is found that the GA approach multicasting routing algorithm provides the best performances in all four networks.

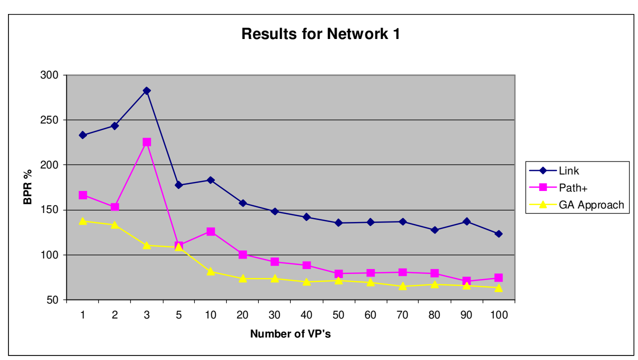


Fig. 3a: <u>Results for Network 1</u>

Fig. 3b: Results for Network 2

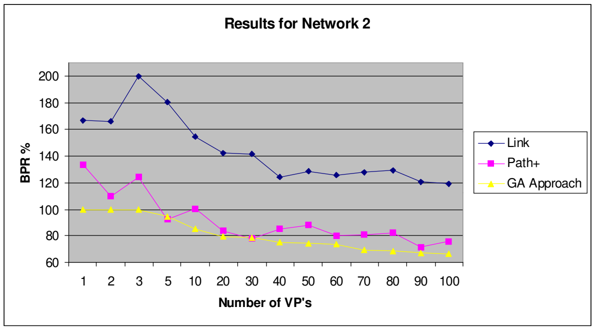


Fig. 3c: Results for Network 3

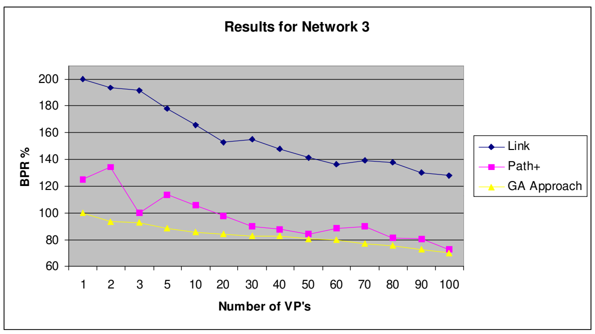
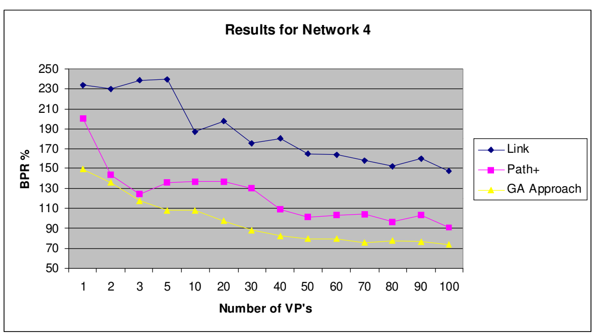


Fig. 3d: Results for Network 4



6 CONCLUSIONS

In this paper, we have proposed a fault tolerant routing algorithm for ATM network. Instead of following the majority of the past researches, we use Genetic Algorithm rather than Linear/Integer Programming as an optimization methodology. Based on a set of input customer traffic demands, the system can produce a set of working and backup VP 's that can satisfy the demands and at the same time minimizing the working and spare capacities and delays. The key contribution of the paper are summarized below:

- The system employs Genetic Algorithm instead of Linear/Integer Programming – this is a breakthrough from the past researches.
- The system can tackle both multicast or unicast traffic demand.
- It can produce node- or link- disjointed working and backup VP's
- The system can jointly optimize the working and backup VP 's altogether to achieve a lower amount of spare capacity.
- The system can take a set of pre-defined working VP's and optimizing the backup VP 's for lower spare capacity required – spare optimization.

We have successfully demonstrated that GA is a good alternative to LP / IP on the Self-Healing Network Design problem. We believe that our study has paved the way for the future followers/researchers to explore and employ GA on the Self-Healing Network Design problem. We believe for GA's powerful capabilities, it will completely eliminate the use of LP / IP on this area.

References

Shanzhi Chen, Shiduan Cheng, Bin Chen, and Junliang Chen, "An Efficient Spare Capacity Allocation Strategy for ATM Survivable Networks", *GLOBECOM 1996*, p.442-446.

Lawerence Davis, David Orvosh, Anthony Cox and Yuping Qiu, "A Genetic Algorithm for Survivable Network Design", *Proceedings of the Fifth International Conference on Genetic Algorithm*, Urbana-Champaign, July 1993, pp.408-414

Berna Dengiz, Fulya Altiparmak and Alica E Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks", *IEEE transactions on Evolutionary Computation*, vol. 1, no.3, Sept., 1997, pp.179-188

Rainer R. Iraschko, M.H. MacGregor, and Wayne D. Grover, "Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks", *IEEE/ACM Transactions on Networking*, Vol. 6, No. 3, June 1998, p.325-336

King-Tim Ko, Kit-Sang Tang, Cheung-Yau Chan, Kim-Fung Man and Sam Kwong, "Using Genetic Algorithms to Design Mesh Networks", *IEEE Computer*, August 1997, pp.56-61

Kazutaka Murakami and Hyong S. Kim, "Joint Optimization of Capacity and Flow Assignment Based on Line and End-to-End restoration", *IEEE transactions on Networking*, vol.6, no.2, April, 1998, pp.207-221

Yijun Xiong and Lorne G. Mason, "Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks", *IEEE/ACM Transactions on Networking*, Vol. 7, No.1., February 1999. pp.99-110

Qingfu Zhang and Yiu-Wing Leung, "An Orthogonal Genetic Algorithm for Multimedia Multicast Routing", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No.1, April, 1999, p.53-62