Virtual Path Topology design in ATM Networks by using Genetic Algorithm

Ali Kadhum Idrees
Department of Computer Science, College of Sciences for Women, University of Babylon

Abstract:

In this paper, we propose a two-phase design method for Virtual Path (VP) topology static design in ATM networks. The goal in the first phase is to determine VP terminators, capacities and a set of alternate paths for each VP, while in the second phase we try to choose a path for each VP from the corresponding set of alternate paths so that the overall network performance is optimized by using Genetic Algorithm(GA). Fixed-length chromosome has been used for encoding the problem according to the number of VPs. The crossover and mutation together provide a search capability that results in a good quality of solution and enhanced rate of convergence. The simulation results explain that the proposed GA exhibits a much better quality of solution (optimal VP network) and a much higher rate of convergence.

KEYWORDS: virtual path topology design, ATM networks, Genetic Algorithms, Optimization problems, Quasi-static VP control.

تصميم طوبولوجية المسارات الظاهرية في شبكة ATM بواسطة استخدام خوارزمية جينية

الخلاصة

أقترح في هذا البحث، طريقة تصميم بمرحلتين ، لغرض التصميم الثابت لطوبولوجية المسارات الظاهرية في شبكات ATM . الهدف من المرحلة الأولى هو تحديد طرفيات المسارات الظاهرية، السعات ومجموعة المسارات البديلة لكل مسار ظاهري، بينما في المرحلة الثانية فنحاول اختيار مسار لكل مسار ظاهري من مجموعة المسارات البديلة ذات العلاقة لكي يكون أداء الشبكة الإجمالي امثلياً بواسطة خوارزمية جينية. تم استخدام كروموسوم ثابت الطول لترميز المسألة طبقاً لعدد المسارات الظاهرية. يزود كل من التزاوج والطفرة قابلية بحث التي تنتج حل جيد الجودة ونسبة محسنة من التقارب. نتائج المحاكاة تبين أن الخوارزمية الجينية المقترحة تعرض جودة حل أفضل (شبكة مسارات ظاهرية مثلى) ونسبة أعلى من التقارب.

1. Introduction:

The next-generation high-speed network, i.e., the broadband integrated service data network (B-ISDN), is expected to carry diverse services with stringent quality of service (QoS) requirements. In the current Internet, each link is fairly shared by all services, and the packets of a specific session are forwarded at intermediate nodes on a best-effort basis, and thus may arrive at the destination out of order. For this reasons, the current internet architecture is inappropriate for the next-generation networks. That is why connection-oriented architectures such as Asynchronous Transfer Mode (ATM) have been proposed and evaluated. 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 VP terminator pair, physical path and capacity of each VP with the objective of optimizing the overall network performance (Feng, 2001). The VP topology design can be formulated as an optimization problem and can be solved by genetic algorithm. The hardware implementations (e.g., field-programmable gate array (FPGA) chips) of neural networks (NN) or genetic algorithms are extremely fast. They are not very sensitive to network size. The quality of solution returned by NNs is constrained by their inherent characteristics. GAs are flexible in this regard. The quality of solution can be adjusted as a function of population. In addition, NN Hardware is limited in size, it can not accommodate networks of arbitrary size because of its physical limitation. GA hardware on the other hand, scales well to networks that may not even fit within the memory. It is realized by employing parallel GA over several routers. Therefore, GAs (especially hardware implementations) are clearly quite promising in this regard (Ahn et al., 2001; Tufte and Haddow, 1999). Most previous work in the design of a VP topology is based on conventional algorithms. Several VP topology design algorithms have been proposed with different contribution. Gerla et al.(1989) proposes a VP topology design problem was formulated as an optimization problem in which the objective is the minimization of the average packet delay, while the decision variables are the routing of VPs and the allocation of bandwidth of such VPs. Since the objective function was non-covex, the proposed downhill technique could not guarantee that the global minimum is reachable starting from an arbitrary initial solution. In addition to the static design, the global reconfiguration of VP topology was also investigated. The limitation of this work lies in that some underlying requirements such as the limited number of VPIs were not taken into account. Chlamtac et al. (1993) took a relatively comprehensive consideration on the VP topology design problem. They separated the design procedure into three phases, i.e., determining VP terminators, handling capacity constraints, and optimizing VP routes. Approximate algorithms for each phase were also provided, but performance analyses and experimental results of these algorithms were not given. Moreover, the proposed objective functions lack clear physical meanings. Ahn et al. (1994) pointed out that it is impossible to establish direct VPs between all users due to the inadequacy of VPIs especially in the case of large-scale networks. They suggested dividing the VP network into multiple sub-networks, with each responsible for certain type of services and only using a portion of the VPIs. Based on this idea, the design problem was decomposed into traffic flow optimization problem and a VP-layout generation problem. There was no clear optimization objective, and only some constraints such as time delay and link capacity were considered in the design. The study of Krishnan and Cardwell (1994) showed that the restriction that VCCs only use single-VP or two-VP routes, which simplifies the routing problems, imposes no penalty on the network cost when the offered loads are large enough to ensure moderate connectivity, and when VP bandwidth is allocated in units of the bandwidth of the dominant traffic in the network. By considering the VP topology design and VC routing problem jointly, Cheng and Lin (1994) formulated an optimization problem in which the objective was to minimize the total call blocking rate, and the decision variables are VP terminators, routes and capacities, and VC routes. Since this problem is NP-hard, approximation were utilize , resulting in a simplified formulation, which were then solved using an iterative algorithm. Kim (1995) upheld that, in the VP topology design, the tradeoff between the increased capacity costs and the reduced control costs should be explicitly reflected in the objective function. Following this criterion, they constructed an objective function containing the two types of costs, as well as the buffer cost. In their problem setting, only direct VPs were allowed, and decision variables were VP routes and capacities. In the work conducted by Gerstel et al. (1996), a heuristic method was proposed to determine VP terminators and the routing of VCCs over the VP network. They first presented a scheme for designing the VP layout required to support VCCs between a specific node and any other nodes (which is called a restricted case in), and then extended this scheme to more complicated cases. Corresponding strategies were given for tree networks, k-separable networks, meshes and general networks. In the work carried out by Farago et al. (1999), the theory of random graphs was employed to design the VP network topology, or more precisely, to determine the VP terminators. They proved that the optimality vs. scalability dilemma could be resolved by means of the proposed algorithm.

In this paper, we proposes a two-phase design method for optimal Virtual Path(VP) topology static design in ATM networks. The goal in the first phase is to determine VP terminators, capacities and a set of alternate paths for each VP, while in the second phase we try to choose a path for each VP from the corresponding set of alternate paths so that the overall network performance is minimized by using Genetic Algorithm(GA).fixed-length chromosome has been used for encoding the problem according to the number of VPs.The crossover and mutation together provide a search capability that results in a good quality of solution and enhanced rate of convergence.

This paper is organized as follows, the VP topology design and control in ATM networks is described in section 2, section 3 describes the GA for optimization, in section 4, the two-phase static design of VP topology is described, in section 5, the experimental results are shown, the conclusion and future work of this paper is described in section 6.

2. The Virtual Path topology design and control in ATM networks:

2.1. Basic concepts (Perros, 2005; Grover, 2003; Pandya and Sen, 1998; Bannister et al.,2004)**:**

ATM is designed to carry both real-time and non-real-time applications in a single network. It is connection-oriented to ensure that the cells for a specific application can reach the destination in order. The ATM packet is known as cell, and it has a fixed size of 53 bytes long which makes the use of regular switching architectures possible, and the delay at each node is more predictable in comparison with the case where variable-sized packets are used. It consists of a payload of 48 bytes and a header of 5 bytes. Two different formats for the cell header were adopted, one for the user network interface (UNI) and a slightly different one for the network-network interface (NNI). The UNI is concerned with the interface between an ATM end device and the ATM switch to which it is attached. An ATM end device is any device that can be attached directly to an ATM network and that can transmit and receive ATM cells. The NNI is used between two ATM switches belonging to the same network or to two different networks.

ATM support five different service classes with each providing a different level of QoS. The class of service is specified by the source during connection set-up. The five services are defined by the ATM Forum as follow:

- Constant Bit Rate (CBR) service, which is similar to a leased line is characterized by the peak cell rate. It requires guaranteed bandwidth and delay from the network.
- Variable Bit Rate (VBR) service, which is described by the peak cell rate and the upper bound of the average cell rate. It is divided into real-time VBR (rt-VBR) and non-real-time VBR (nrt-VBR).
- Available Bit Rate (ABR) service, which is designed for data applications, uses rate-based congestion control to manage the traffic in the network. In case of network congestion, the resource management (RM) cells are used to inform the sources to reduce the data rate.
- Guaranteed Frame Rate (GFR) service, With GFR, a bandwidth level is defined such that traffic is guaranteed not to fall below this minimum bandwidth. The traffic may receive a performance above this bandwidth, but at a best-effort performance level.
- Unspecified Bit Rate (UBR) service, which does not provide the user with any bandwidth or delay guarantee. The network just tries its best to carry such traffic.

There are two types of connection in ATM networks, Virtual Path Connection (VPC) and Virtual Channel Connection (VCC). A VPC is a logical connection between two nodes, which are termed VP terminators. a VPC may contain a bundle of VCCs, where as a VCC may be a concatenation of several VPCs. VPCs and VCCs are identified by the VP identifier (VPI) and VC identifier (VCI) in the ATM cell header, respectively. Over the physical network, the VP topology, or VP network can be established and updated freely.

2.2. VP topology control: necessities and strategies (Feng, 2001; Ahn et al.,1994; Burgin and Dorman, 1991; Gerstel and Segall,1995):

The VP mechanism was proposed to facilitate the resource management. This is due to the consideration that the network resources must be used very efficiently since in ATM networks there may exist a variety of services with diverse statistical characteristics and a wide range of information bit rates. The significance of the VP concept lies in that it permits VCCs to be handled in groups, resulting in substantially reduced nodal processing costs and a simplified network architecture. However, a drawback also ensues from the use of this mechanism since the efficiency of capacity utilization is jeopardized due to the decrease of the statistical multiplexing capability. Thus, great care must be taken to resolve this discrepancy.

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 physical routes, to the best state. Previously proposed VP control methods can be classified into three categories, dynamic VP control, static VP control, and quasi-static VP control. Dynamic VP control tries to promptly adjust the VP topology to keep its responsiveness to the instantaneous change of traffic flows. The first glance at this method may make one think that it can dramatically improve the network throughput, but it suffers from an insurmountable drawback that the processing costs may considerably increase, which may even offset the advantages provided by the VP mechanism. In the other extreme where the VPs are determined statically, i.e., the VP capacities and routes are fixed during the lifetime of a network, processing costs may be minimized at the expense of the probably incessant decrease of the throughput. Therefore, 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 the above reasons, the quasi-static VP control is so far regarded as the best strategy. In this method, the tasks of VP topology design and control are accomplished by three operations, i.e., static optimization, local modification, and global reconfiguration. If we partition the procedure of VP topology design and control into static design stage and dynamic maintenance stage, then the first operation, i.e., static optimization occurs in the static design stage, while the local modification and global reconfiguration take place in the dynamic maintenance stage. The temporal relation of the three operations is illustrated in figure 3. for specific network, the local modification in effect spans the entire dynamic maintenance stage, while the global modification occurs at regular intervals or when the monitored performance measures exceed specified thresholds. In the static design stage, the aim is to optimally design the VP topology based on the traffic parameters declared by users, ensuring that the traffic requirements are 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 such as the capacity, route, and VPT pair of each VP is not retained in the central node but is distributed to associated local nodes as well. During the local modification, 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 and provided by the attached users. Adjustment can be made only on those associated VPs to track the instantaneous change of traffic. Therefore, simple algorithms are preferred to guarantee their responsiveness.

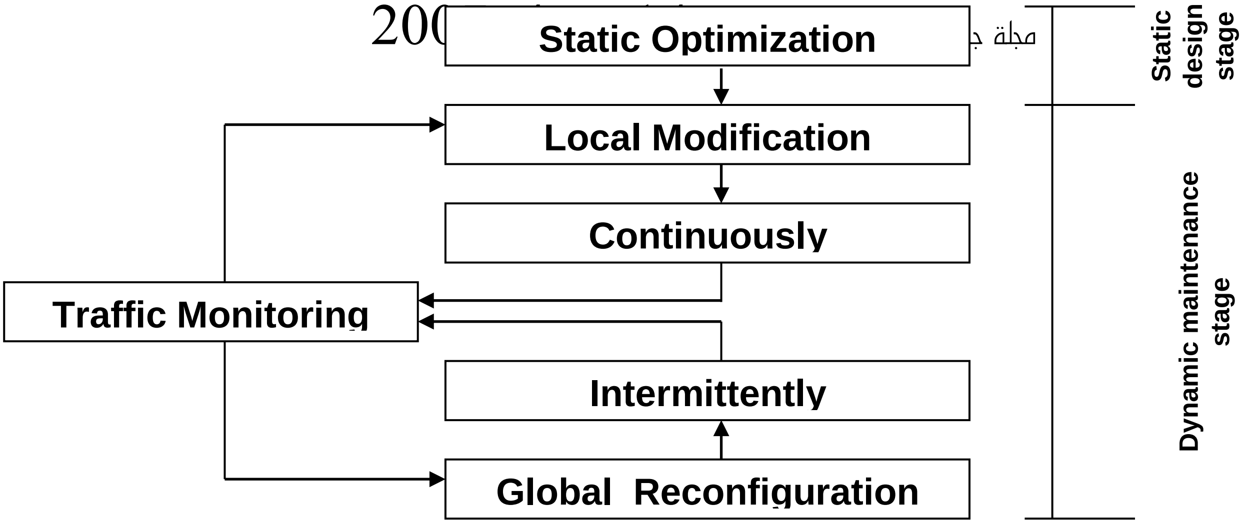

Figure (3): Quasi-static VP control

The modifications on VPs may include establishing new VPs, rerouting or tear 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 for setting up new VPs and also is necessary for the global reconfiguration. The global reconfiguration operation should result in maximized network performance, which may have been aggravated after a period of local operations.

The salient features of the quasi-static 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 the local nodes may be alleviated and well-balanced. The amount of information interchanged between the central node and the local nodes can be restricted 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.

3- The Genetic Algorithms for optimization:

The GAs are search algorithms that simulate the process of natural selection and survival of the fittest. During the search process, it can automatically achieve and accumulate the knowledge about the search space, and adaptively control the search process to reach the overall optimal solution(Goldberg, 1989; Hourani, 2004). GAs are capable of handling complex and irregular solution spaces, and they have been successfully applied to various difficult optimization problems (Michalewicz, 1996; Chambers, 2001). In the quasi-static VP control method, the problem that arises either in the static optimization or in the global reconfiguration can be formulated as a common combinatorial optimization problem, in which the final goal is to find an optimized VP network according to a specific objective function, and we used the GA to solve this optimization problem.

4- The two-phase 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. Based on previous research results, we herein present a two-phase design method for the VP topology static design problem. The goal in the first phase is to determine VP terminators, capacities, and a set of alternate paths for each VP, while in the second phase we try to choose a path for each VP from the corresponding set of alternate paths by using a genetic algorithm so that the overall network performance is optimized. The problem setting in the second phase is the same as that in the global reconfiguration, and correspondingly can be formulated as the same Virtual Path Topology Design problem, which will he discussed later.

4.1 problem and solutions of Phase I :

4.1.1 Determining the terminators and quantity of VPs:

The determination of the quantity and terminators of VPs is closely related to the limited number of hits 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 UNI cell header and 12 bits in a NNI cell header. For this reason. the amount of VPIs is excessive in some cases, while inadequate in some other cases.

For a network of a 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 VPIs are available. For instance, we may assign a VP to transmit each service so that the QoS can be strictly controlled.

For a large networks, we can reduce the number of VPs so that the VP network can remain fully interconnected by direct VPs. There are two methods can be used to reduce the number of VPs (Gerla et al., 1989). 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. The reason to do so lies in that the CBR and VBR services have similar stringent QoS requirements, and using a separate VP to transmit ABR/UBR services can avoid their adverse influence on CBR/VBR services. Second, we may design a new VP switch which can identify the physical link ID, or port ID (PID). In doing so, for an ATM cell incident to a switch, the switch must check both of its VPI and PID to determine its outgoing path. In fact, this technique is equivalent to an expansion of the number of bits for VPI in the cell header. For such a switch connected to K physical links, a total number of K * 4096 VPs can be accommodated. For every large scale networks, or when there is a difficulty in realizing VP switches capable of identifying PIDs, 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.

4.1.2 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 is based on the consideration that we can perform further adjustments on the VP capacities after the second phase of static design. Moreover, during the local modification of the quasi-static 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. 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.

4.1.3 Determining the alternate routes:

In order to establish a VP between a SD pair, what physical route should be chosen to accommodate it? we used a heuristic method to choose the alternate routes. This method has two basic rules (Feng, 2001): (1) choose the shortest routes (in terms of hop count), (2) any two alternate routes between a SD pair should cross as small a number of common nodes as possible. The choice of the shortest routes can reduce the amount of total capacity to he used by the corresponding VP. The second rule is due to the security considerations. If all alternate routes for a given Source-Destination(SD) pair cross the same link and unfortunately that link is broken, then no alternate route can accommodate the VP. To obtain such alternate routes, the K-shortest path algorithm or the recursive algorithm provided in (Chambers, 2001) can be used to first find a number of routes, and then choose the best candidates.

4.2 Global Reconfiguration: VP Topology Optimization:

The global reconfiguration and the problem setting in the second phase of the static design can be formulated as a common Virtual Path Topology Design Optimization problem and we can use the genetic algorithm to solve it. The Virtual Path Topology Design problem assumes that, the following are known: the physical topology, the available capacity of each link. VP terminators, and the bandwidth and a set of alternate routes for each VP. The goal is to select a route for each VP from its alternate routes guaranteeing that the resulting VP topology is optimized according to a specific objective function.

4.3 Virtual Path Topology Design problem based on Genetic Algorithm:

In this subsection we propose a method based on the Genetic Algorithm to solve the VP Topology optimization problem. The GA used to select exactly one route from a number of alternate routes to accommodate the VP for each SD pair. The GA will produce optimal or near optimal solution for VP topology design problem. We used the following algorithm in this paper:

```
Algorithm VirtualPathTopologyDesign_GA
   Initialization ( Population )
   Evaluation ( Population )
   Generation ← 0
    Do While ( Not Stop-Criteria )
        Selection ( Population, Parents )
        Crossover ( Parents, Offspring , Pc )
        Mutation ( Offspring, Pm )
        Evaluation (Offspring )
        Replacement ( Population, Offspring)
        Generation ← Generation +1
    Loop
    SelectOptimalVirtualPathTopologyDesign ( Population )
End VirtualPathTopologyDesign_GA
```

The detail of the above algorithm is explained as follow:

1- the first step performed by the GA is to initialize a population of the individuals, each individual consists of a chromosome and a fitness value. The chromosome is encoded with integer encoding. The length of the chromosome equal to M, where M is the total number of virtual paths (VPs) that needed to be established in the ATM network. Each gene in the chromosome will contain an integer value that represent the sequence number of the alternate route in the alternate route set that belong to the appropriate Source-Destination (SD) pair. Each gene i willed with a random value in the range [1..Ri], where, Ri represent the number of alternate route for the ith VP. Each SD-pair in the ATM network will have the best four alternate route that calculated in the phase I. the chromosome consists of M VP and each VP represent a SD-pair that needs only one route that accommodate the VP.

2- Evaluation of the individuals: each individual is evaluated by using fitness function. The network throughput maximization is chosen to be the objective function for the GA-based VP topology design problem because the maximization of the network throughput is the most representative goal for the network performance (Feng, 2001). This include the minimization of the summation of all link capacity utilization ratios (the ratio of the offered capacity to the total capacity). Furthermore, to ensure that the load is network-wide balanced, the capacity utilization ratio of each link should be approximately the same. Thus, we may take into account another objective, i.e., the minimization of the square summation of all link capacity utilization ratios. There is another important factor should be also taken into account, the processing load or throughput of each node. Since two distinct nodes are generally connected to different links, the amount of information passing through them is also different. This leads to two another similar objectives: the minimization of the summation of all nodal capacity utilization ratios, and the minimization of the square summation of all nodal capacity utilization ratios. The nodal capacity utilization ratio is defined as the ratio of the amount of information traversing the node to the maximal throughput of the node. We can calculate the fitness of each individual from the following relation:

$$Min\ O_{a_i} = \sum_{k=1}^{L+N} \mu_k + \beta_k \mu_k^2 \qquad (1)$$

$$subject\ to\ \mu_k \le 1,\ k = 1,2,\ldots,L+N$$

$$\mu_k = \begin{cases} \hat{\mu}_k & \text{If} \quad 1 \le k \le L \\ \tilde{\mu}_k & \text{If} \quad L+1 \le k \le L+N \end{cases}$$

$$\beta_k = \begin{cases} \hat{\beta}_k & \text{If} \quad 1 \le k \le L \\ \tilde{\beta}_k & \text{If} \quad L+1 \le k \le L+N \end{cases}$$

$$\hat{\mu}_k = \frac{\hat{f}_k}{\hat{C}_k} \quad , \quad \tilde{\mu}_k = \frac{\tilde{f}_k}{\tilde{C}_k} \quad , \quad \hat{\alpha}_k = \frac{1}{1-\hat{\mu}_k} \quad , \quad \tilde{\alpha}_k = \frac{1}{1-\tilde{\mu}_k}$$

We can express the amount the information passing through the kth link or node as follow:

$$f_k = \sum_{i}^{M} \sum_{j}^{R_i} B_i T_{ijk} \quad , \quad k = 1,2,\ldots,L+N \qquad (2)$$

$$T_{ijk} = \begin{cases} \hat{T}_{ijk} & \text{If} \quad 1 \le k \le L \\ \tilde{T}_{ijk} & \text{If} \quad L+1 \le k \le L+N \end{cases}$$

$$C_k = \begin{cases} \hat{C}_k & \text{If} \quad 1 \le k \le L \\ \tilde{C}_k & \text{If} \quad L+1 \le k \le L+N \end{cases}$$

The fitness is computed as follow:

$$fitness = \frac{1}{\sum_{i=1}^{M} O_{a_i}} \qquad (3)$$

where:

M: the total number of VPs.

R_i : the number of alternate routes for the ith VP.

B_i : the bandwidth of the ith VP (bits/s).

$\hat{T}_{ijk}$: equal to 1 if route j of VP i goes through link k and 0 otherwise.

$\tilde{T}_{ijk}$: equal to 1 if route j of VP i goes through node k and 0 otherwise.

O_{a_i} : the value of objective function after applying the relation (1) on gene i that has the alternate route its sequence number a_i.

L: the total number of physical links.

N: the total number of physical nodes.

$\hat{C}_k$: the capacity of link k (bits/s). $\tilde{C}_k$: the capacity of node k (bits/s).

$\hat{f}_k$: the information bit rate on link k (bits/s). $\tilde{f}_k$: the information bit rate on node k (bits/s).

$\hat{\mu}_k$: the capacity utilization ratio of link k. $\tilde{\mu}_k$: the capacity utilization ratio of node k.

3- The Selection:

In this step, the GA will select two parents for crossover and to produce new offspring. We used in this paper the binary tournament selection (BTS), where, two different individuals are selected randomly. The individual that has higher fitness will win to be added to the crossover mate. If the fitness value of the first individual is equal to the fitness value of the second individual, one of them will be selected randomly (Schmidt and Stidsen, 1997; Michalewicz, 1996).

4- The Crossover:

it is the process of exchanging the parents genes to produce one or two offspring that carry inherent genes from both parents. It occurs with a certain crossover probability Pc, otherwise, the two parents are copied as offspring. We used in this paper three crossover method (Goldberg, 1989; Michalewicz, 1996; Ashlock, 2006) such as:

- Single point crossover (1X): it operates by picking a random point within the two parent chromosomes, and then exchanging the genes of the two chromosomes at or below this point to produce two new offspring.
- Two point crossover (2X): it operates by picking two random points within two parent chromosomes, and then exchanging the genes between these points in each parent to produce two new offspring.
- Uniform crossover (UX): genes are randomly copied from the two parents to compose two new offspring with a probability Pu (Pu=0.5).

5- The Mutation: it take place after performing the crossover. It randomly changes a selected gene in the offspring. This occurs at an assigned rate, usually very infrequently. The purpose of mutation is to prevent falling into local optimal solution of the solved problem (Hourani, 2004). We used in this paper, three mutation method such as:

- One point mutation (1M): it randomly select a one gene in the chromosome and then changes it's content with a randomly selected value within the range [1.. R_i].
- Two point mutation (2M): it randomly select a two different genes within the chromosome, and then each of them replaces it's content with a randomly selected value from the range [1.. R_i].
- Exchange mutation (EX): it randomly select a two different genes within the chromosome, and then exchange the content of the first gene with the second gene.

6- Offspring evaluation: the new offspring is evaluated by using the formula (3) in step 2, in order to calculate the fitness value of the new individual.

7- The Replacement: it replaces the old individual that has lower fitness with new individual that has a better fitness. We used in this paper, a Triple Tournament Replacement (TTR) by selecting randomly three different individual from the population, the new offspring will be replaced with the worst one of the three selected individuals, if its fitness larger than the worst one, otherwise, the new offspring don't replaced (Schmidt and Stidsen, 1997; Goldberg, 1989).

8- The termination criterion: we used the On-Line performance as a termination criterion to measure the convergence of GA. The following algorithm explain the termination criterion:

```
Algorithm Termination-Criterion
    New-Avg ← 0
    Generation ← 1
     Flag ←  0
     Sum-Fit ← 0
     Do While (Generation <  MaxGen ) And (Flag <  Max-count)
         Old-Avg ← new-Avg
         ( Selection, Crossover, Mutation, evaluation, replacement)
         Sum-Fit ← Sum-Fit + Child.Fit
          New-Avg ← Sum-Fit / Generation
          If  | New-Avg – Old-Avg | < Difference then
              Flag ← flag + 1
              Else Flag ← 0
          EndIf
          Generation ← Generation +1
        Loop
  End of Termination-Criterion
```

Where the New-Avg is the current average fitness, Old-Avg is the old average fitness, Difference is the permitted difference, Chil.Fit is the new child fitness, Max-Count is the maximum number for the difference in the fitness (Max-Count = 5), Sum-Fit is the fitness sum, and Generation is the current generation number.

9- Select the optimal solution: after the GA converged to the optimal solution, we will select the optimal solution that represent the optimal VP design for the ATM network.

5- The Results:

in this section, the proposed GA-based virtual path topology design algorithm is applied to a network to test it's performance. The network model includes 10 nodes and 36 links that assumed to be an ATM network as explained in figure 4. we assume that the six links that connecting to node 10 each have a capacity of 50 Kbps, while any other link has a capacity of 38.4 Kpbs. In addition, we assume that the maximal throughput of each node is 150 Kbps. The source-destination nodes and the traffic requirement between each VP are enumerated in table 1.

Table (1): traffic requirements for VPs (cells/second)

VP #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Source	1	1	1	2	2	2	3	3	3	3	4	4	4	4	5
Destination	6	8	9	4	6	8	6	7	8	9	2	5	8	9	1
traffic	5	10	5	5	8	14	13	5	5	20	5	5	10	5	5

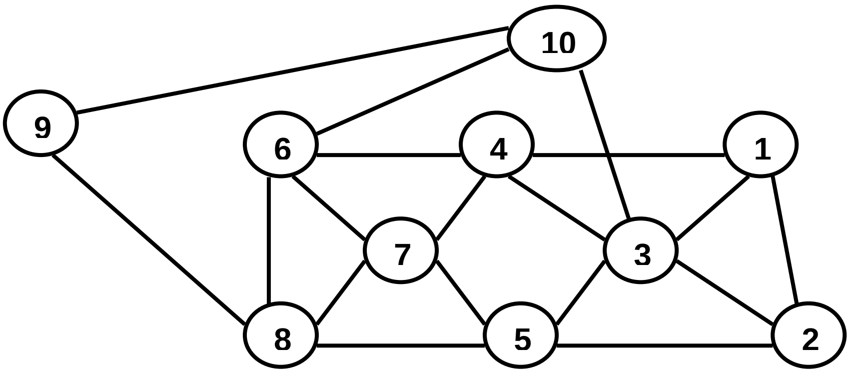

Figure (4): ATM network model.

5.1 Determine the alternate routes:

by means of the algorithm proposed in (Feng, 2001), we obtain 4 alternate routes for each VP as listed in table 2.

Table (2): Alternate routes of VPs

VP #	Source	Destination	Intermediate nodes of alternate routes			
			Route 1	Route 2	Route 3	Route 4
1	1	6	3-4	3-10	4	4-7
2	1	8	2-5	3-5	4-6	4-7
3	1	9	2-3-10	3-10	4-3-10	4-6-8
4	2	4	1-3	1	3-1	3
5	2	6	1-4	3-10	5-7	5-8
6	2	8	3-5	3-10-9	5-7	5
7	3	6	1-4	4	4-7	10
8	3	7	2-5	4	5	10-6
9	3	8	2-5	5	10-6	10-9
10	3	9	4-6-10	5-8	10-6-8	10
11	4	2	1	1-3	3-1	3
12	4	5	1-2	3-2	3	7
13	4	8	6-7	6	7-6	7
14	4	9	3-10	6-8	6-10	7-8
15	5	1	2	2-3	3	7-4

5.2 **The effect of the Pc and Pm on the AvgGen and VPTD optimality**:

In this experiment, we study the effect of the probability of crossover Pc and the probability of mutation Pm on the Average number of Generation (AvgGen) and the Virtual Path Topology Design (VPTD) optimality. We use the following relations to calculate the AvgGen and VPTD optimality.

$$AvgGen = \frac{\sum_{i=1}^{10} Gen_i}{10} \quad , \quad VPTD\ optimality = \frac{10 - No - Opt}{10} * 100$$

where, Gen_i is the number of generation in the i consecutive run.

$No - Opt$: the number of not or near optimal virtual path topology design.

We set the population size (PopSize) to 30, M=5, where the GA will uses the first five VPs in table(2) that have traffic requirements as in table(1). MaxGen=700, R_i =4,and Difference=0.0005. The GA used the following operators: BTS,1X,1M, and TTR in this experiment. Table (3) explain the effect of Pc and Pm on AvgGen and VPTD optimality.

Table (3): the effect of Pc and Pm on AvgGen and VPTD optimality

		Probability of Mutation (Pm)									
		Pm= 0.01		Pm= 0.1		Pm= 0.2		Pm= 0.3		Pm= 0.4	
		AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)
Probability of Crossover (Pc)	0.1	110.6	20	109.1	60	109.8	50	109.6	50	112.2	90
	0.2	109.8	20	110.7	60	110.7	60	109.8	50	112.4	90
	0.3	108.7	20	111.3	60	112	70	110.9	60	110.7	90
	0.4	111.1	30	112.9	70	111.8	90	111.7	90	113.2	100
	0.5	111.2	50	112.6	60	111.4	70	113.1	90	113.1	80
	0.6	112.2	40	112.6	80	113.1	80	113	70	112.8	100
	0.7	110.6	50	112.5	70	113.2	80	**112.6**	**100**	113.2	100
	0.8	113	70	112.6	70	114	100	112.9	100	113.4	100
	0.9	113	40	113.1	70	113.5	80	113.8	90	113.3	100

From simulation results, we see when the Pc=0.7 and Pm=0.3, this leads to less generation number with optimal virtual path topology design.

5.3 The effect of the PopSize on AvgGen and VPTD optimality:

In this experiment, we study the effect of the population size (PopSize) on the Average number of Generation (AvgGen) and the Virtual Path Topology Design (VPTD) optimality. We set the M=5, where the GA will uses the first five VPs in table(2) that have traffic requirements as in table(1). MaxGen=700, R_i =4,and Difference=0.0005. The GA used the following operators: BTS,1X,1M, and TTR in this experiment. Figure 5 explain the effect of Population size on the AvgGen and VPTD optimality.

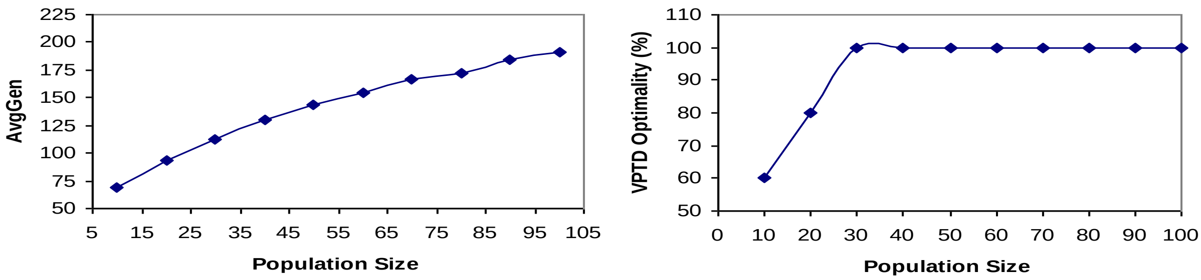

Figure (5): The effect of Population size on the AvgGen and VPTD optimality

From simulation results, we see when the population size increase, this leads to increase each of the number of generations and the virtual path topology design optimality. We make a good balance between the AvgGen and VPTD optimality by taking the PopSize equal to 30.

5.4 The effect of the Crossover and Mutation methods on AvgGen and VPTD optimality:

In this experiment, we study the effect of the Crossover and Mutation methods on the Average number of Generation (AvgGen) and the Virtual Path Topology Design (VPTD) optimality. We set the M=5, where the GA will uses the first five VPs in table(2) that have traffic requirements as in table(1). MaxGen=700, R_i =4,and Difference=0.0005. table 4 explains the effect of the crossover and mutation methods on AvgGen and VPTD optimality

Table(4): the effect of the crossover and mutation methods on AvgGen and VPTD optimality

	1X		2X		UX	
	AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)	AvgGen	VPTD optimality (%)
1M	112.6	100	**112.6**	**100**	113.9	100
2M	112.1	70	112.6	60	113.5	90
EM	112.7	60	113.4	70	113.5	80

From simulation results, we see when the GA uses the 2X and 1M methods will give less number of generations and 100% virtual path topology design optimality.

5.5 The effect of the number of VP that need to be established on AvgGen:

In this experiment, we study The effect of the number of VP that need to be established by the ATM network on the Average number of Generation (AvgGen). This experiment also explain the routes that selected by the GA for each VP. The GA will uses the VPs in table(2) that have traffic requirements as in table(1). MaxGen=700, and R_i =4. The GA uses the BTS, 2X, 1M, and TRR methods in this study. The term Route Nos. denotes the routes assigned to the virtual paths. More clearly, in a certain Route Nos., ith number j denote the jth alternative route of the ith virtual path and this route is assigned to this VP. Table (5) explain the effect of VP number that need to be established on AvgGen.

Table(5):The effect of the number of VP that need to be established on AvgGen

Virtual Path Number	AvgGen	Route Nos.	Difference
3	23.4	211	0.03
6	163	242133	0.0001
9	348	222123434	0.00001
12	402	444233333114	0.0000025
15	650.8	334213232114131	0.0000021

From simulation results, we see when the number of virtual path increase, this lead to increase the number of generation of GA to converge to optimal VP topology design with decrease the Difference parameter that used by the Stopping criteria of the GA. We see also in the Route Nos. field the results that obtained by the GA with a different number of VP that need to be established by the ATM network. Each number j refer to alternative route that will accommodate a certain i VP (SD-pair). will give less number of generations and 100% virtual path topology design optimality.

6. Conclusion and Future work:

In this paper, the quasi-static virtual path control scheme is elaborated. A detailed analysis demonstrates that this scheme has many advantages such as feasibility and flexibility. We provide a GA-based approach to solve the virtual path topology design optimization problem arising in the static design and global reconfiguration stages of this scheme. The hardware implemented GA (e.g., field-programmable gate array (FPGA) chips) can produce a solution in a very short time. The simulation results explain that the proposed GA can produce optimal virtual path topology design and its simplicity and flexibility of establishing any number of virtual paths in ATM network. In our algorithm, we assume there is only one VP between each pair of VP terminators. In fact, the assumption can be easily extended to the case where multiple virtual paths are to be deployed between any VP terminators in future work.

References:

Ahn, C. W., Ramakrishna, R. S., Kang, C. G., and Choi, I. C. (2001). shortest path routing algorithm using Hopfeild neural network, Electron. Lett, vol. 37, no. 19, pp. 1176-1178.

Ahn, S., Tsang, R. P., Tong, S. R., and Du, D. H. C.(1994). Virtual path layout design on ATM networks. In IEEE Infocom '94, p.p. 192-200.

Ashlock, D. (2006). Evolutionary Computation for Modeling and Optimization, Springer.

Bannister, J., Mather, P. and Coope, S. (2004). Convergence Technologies for 3G Networks :IP, UMTS, EGPRS and ATM, John Wiley & Sons Ltd.

Burgin, J. and Dorman, D. (1991) Broadband ISDN resource management: the role of virtual paths", IEEE communication magazine.

Chambers, L. D. (2001). The practical handbook of genetic algorithms, applications, 2nd edition, Chapman & Hall/CRC.

Cheng, K. T. and Lin, F.Y.-S. (1994). On the Joint Virtual Path Assignment and Virtual Circuit Routing Problem in ATM Networks, Proc. GLOBECOM '94, pp. 777–82.

Chlamtac, I., Farago, A., and Zhang, T. (1993). How to Establish and Utilize Virtual Paths in ATM Networks," Proc. ICC '93, pp. 1368–72.

Farago, A. , Chlamtac, I., and Basagni, S. (1999). virtual path network topology optimization using random graphs, IEEE INFOCOM'99, pp. 491–496.

Feng, G. (2001). neural network and algorithmic methods for solving routing problems in high-speed networks, PHD thesis, university of Miami.

Gerstel, O. and Segall, A. (1995). Dynamic maintenance of the virtual path layout, proc. INFOCOM'95, pp. 330-337.

Gerstel, O., Cidon, I., and Zaks, S. (1996). Optimal Virtual Path Layout in ATM Networks with Shared Routing Table Switches, Chicago Journal of Theoretical Computer Science, The MIT Press,Vol. 1996, Article 3.

Gerla, M., Monteiro, J. A. S., and Pazos, R. (1989).Topology Design and Bandwidth Allocation in ATM Nets," IEEE JSAC, vol. 7, no. 8, pp. 1253–62.

Goldberg, D. E. (1989). genetic algorithms in search, optimization and machine learning, Addison-Wesley.

Grover, W. D. (2003). Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking, Prentice Hall PTR.

Hourani, M. (2004). genetic algorithm for continuous variable optimization with applications to queuing network and gene-clustering problems, MSc thesis, Faculty of Miami university, Oxford, Ohio.

Kim, S.-B. (1995). An Optimal Establishment of Virtual Path Connections for ATM Networks, Proc. INFOCOM '95, pp. 72–79.

Krishnan, K. R. and Cardwell, R. H. (1994). Routing and Virtual-Path Design in ATM Network," Proc. GLOBECOM '94, pp. 765–69.

Michalewicz, Z. (1996). Genetic algorithms + data structures = evolution programs, 3rd edition, Springer- Verlag.

Pandya, A. S. and Sen, E. (1998). ATM Technology for Broadband Telecommunications Networks, CRC Press.

Perros, H. G. (2005). Connection-oriented Networks SONET/SDH, ATM, MPLS and OPTICAL NETWORKS, John Wiley & Sons Ltd.

Schmidt, M. and Stidsen, T. (1997). hybrid systems: genetic algorithms, neural networks and fuzzy logic, DAIMIIR.

Tufte, G. and Haddow, P. C. (1999). prototyping a GA pipeline for complete hardware evolution, in proc. 1st NASA/DOD workshop on evolvable hardware, pp. 76-84.