The CSI Journal on
Computer Science and Engineering
Vol. 6, No. 2 & 4 (b), 2008
Pages 6-14
Regular Paper

Development in Handling the Problem of Fault Recovery in MPLS Networks Using Neural Network

Arash Dana

Department of Electrical Engineering, Islamic Azad University-Central Tehran Branch, Tehran, Iran

Abstract

Rapidly increasing volume of traffic carried by the Internet together with imposing requirements for reliability, quality of service (QoS) and manageability, force the network technology designers to come up with new approaches and solutions. As the Internet moves towards an IP over WDM model, existing means of network engineering to provide assured bandwidth, quality of service and fault tolerance should be substituted. Multiprotocol label Switching (MPLS) has emerged as a technology that can provide many of the functionalities now associated with ATM and/or SONET / SDH without incurring much of the overhead.

MPLS appears to be a suitable place to provide fault tolerance. It is the lowest layer with the knowledge of the entire network topology as well as a point with the necessary traffic engineering capabilities. MPLS recovery mechanisms are applying in data communication network because they can guarantee fast restoration and high QoS assurance. Their main advantages is that their backup paths are established in advance, before a failure event takes place. Most researches on the establishment of primary and backup paths have focused on minimizing the added capacity required by the backup paths in the network. In this paper several back up paths are considered for a primary path. When fault occurs, regarding to remain bandwith of backup paths, the traffic of primary path is splitted over the backup paths using neural network(NN). Ultimately the performance comparision between CBR(case-based reasoning) and NN is presented.

Keywords: MPLS, Restoration, Quality of Service, Performance, Case-Based Reasoning, Neural Network.

1. Introduction

Much research attention has been focused on QoS routing. However, the problem of dynamically routing QoS guaranteed paths with restoration has not been studied extensively. This problem is motivated by emerging trends in backbone and transport networks toward fast dynamic provisioning of restorable bandwidth-guaranteed paths. In MPLS [1], Packets are encapsulate at ingress points with labels that are then used to forward the packets along LSPs. These LSPs can be thought of as virtual traffic trunks that carry flow aggregates generated by classifying the packets arriving at the edge or ingress routers of a MPLS network into “forwarding equivalence classes” [1,2].

This classification into flow aggregates combined with explicit routing of bandwidth-guaranteed LSPs, enables service providers to traffic engineer their networks [3] and to dynamically provision bandwidth-guaranteed paths [19]. Recently, proposals have been made to incorporate restoration mechanisms in MPLS [4, 5, 6]. These restoration mechanisms allow the set up of backup Paths, onto which traffic can be quickly redirected up on failure detection.

There is a large body of literature dealing with network planning scheme for routing restorable LSPs. These centralized schemes typically assume that all the LSP requests are available at the time of route computation is too restrictive. We assume that requests for LSP set up arrive one at time. Each request has an ingress node, an egress node, and an associated bandwidth. The objective of the routing algorithm is to compute an active path and back up paths for every request. This scheme of computing the back up path at the time of LSP setup is referred to as the precomputed restoration path approach. This is different from the restoration approach where the backup path is determined at the time of failure. Such a restoration scheme suffers from two major problems.

The first is that the restoration time increases significantly since the restoration path has to be computed at the point of failure. Further, since the backup bandwidth is not reserved a head of time, there may not be sufficient bandwidth to route the backup path at the time of failure. Therefore, in our model, we assume that the active and the backup paths are computed at the time of LSP initiation and if sufficient bandwidth is not available to setup either the active path or the backup path, then the LSP set up request is rejected. The rest of the paper is organized as follows. Section 2 gives a brief background on the restoration mechanisms. Section 3 describes the proposed algorithm. Section 4 provides the experimental results. Section 5 compares the proposed approaches with other schemes and section 6 concludes the paper.

2. Background

Two different techniques for local protection in MPLS networks have been proposed [7]. The one-to-one backup technique [8,9] creates bypass LSPs for each protected service carrying LSP,at each potential point (Link or node) of local repair. The facility backup technique[10]creates a bypass tunnel to protect a potential failure point (link or node) , such that by taking advantage of the MPLS label stacking mechanism, a collection of LSPs with similar backup constraints can be jointly rerouted, over a single bypass tunnel. In general, the one to one backup technique does not scale very well with the number of supported protected LSPs, since the number of bypass tunnels can quickly become very large, not to mention the enormous load on signaling and routing to support these extra tunnels. In addition, for implementing the one-to-one backup technique, either extensive routing extensions are needed to propagate the set of bypass LSPs and their attribute information, resulting in heavy load on the control plane, or the amount of achievable sharing of protection capacity is sacrificed, by limiting the amount of state that is propagated in the routing updates [8], thus requiring large amounts of spare capacity for protection.

In general the protection scheme for MPLS networks can be classified [9,10], based on whether the protection is local (link based) or end-to-end (path based), and whether the backup resources are dedicated or shared. Fast or local reroute mechanisms, outlined earlier, are instances of link based protection. In path based protection, the entire primary service carrying path is backed up by alternate protection paths, such that any failure on the primary path results in its traffic getting rerouted over its protection paths. In path based protection the reroute is done by the end nodes of the path. Compared to link based protection, recovery may be slower in path based protection schemes, partly because failure information has to reach the end nodes before restoration can be initiated, and partly because even a failure of a single link may effect primary paths of many different ingress egress pairs, all of which may initiate path protection in parallel, resulting in high signaling loads and contention for common resources and crank backs. The protection scheme can be further classified as being pre-planned (e . g . SONET) or even driven (dynamic). The later involves computing bypass routes and reserving protection bandwidth at the time when the working path is provisioned.

These schemes rely on heavy signaling to maintain the reservation and to effect the rerouting on the failure of a link. These scheme although very efficient in lowering the over build tend to have longer restoration times. Note that our scheme is based on pre-planning with the only dynamic component coming from topology changes due to which some protection capacities and bypass tunnels may need to be re-provisioned. For pre-planned facility based fast reroute, the main approaches are through the use of rings in mesh topology. Once the set of rings are identified then pre-planned protection schemes as in SONET are employed. In some of these approaches the network is designed in term of rings [11] or by partially using rings [12]. Thus, these schemes are only applicable to constrained topologies. Some of the other protection schemes provide protection by embedding rings in a mesh based topology. In these schemes each link is covered by a cycle leading to a cycle cover for the network [12]. Each of these cycles is provisioned with enough protection capacity to cover the link that belong to it. On the failure of the link the working traffic is rerouted over the protection capacities in the surviving links of the covering cycle. These are two drawbacks of this problem: one the over build can be significant and second it is hard to find the smallest cycle cover of a given network [13]. An improvement to these schemes are those based on the notion of p-cycle [14].

Here the main idea is that a cycle can be used to protect not just the links on the cycle but also the chords of the cycle, thus showing that far fewer rings maybe sufficient for providing full protection. An algorithm to minimize the total spare capacity, based on solving an integer program over all possible cycles is given in [14]. To the best of our knowledge no fast approximation algorithms for this problem are known. An alternative to cycle covers, intended to overcome the difficulty of finding good covers, is to cover every link in a network with exactly two cycles [15]. Non-ring based approaches to link restoration on mesh networks is generalized look-back [16,17], where the main idea is to select a digraph, called the primary, such that the conjugate digraph, called the secondary , can be used to carry backup traffic for any link failure in the primary. [18] considers the problem of finding the minimum cost augmentation of a given primary network, so that the resulting network is capable of supporting link protection under single link failures, for a given set of links. [25] consider LSP partial spatial protection for connections with various protection grades in MPLS over WDM networks that develops a formulation for this problem.

3. Proposed Approach

3.1. Problem Definition

This paper proposes an efficient rerouting mechanism to establish a LSP along the least-cost recovery paths of all possible alternative paths that can be found on a working paths. Reservation of spare bandwidth on a backup path causes an additional cost and also degrades network efficiency [22]. Shared backup path protection is one of the most commonly adopted approaches in current communication carriers for achieving 100 percent restorability in the single failure scenario and has caught extensive interest from both industry and the academia[26]. The method for splitting the traffic of faulty LSP, in our approach, is based on the idea that no band-width should be Pre-reserved on the Pre-assigned protection LSPs[23].MPLS fast reroute achieves rapid restoration by computing and signaling backup label switched path tunnels in advance and re-directing traffic as close to failure point as possible[24].

This is because the same band-width may at the same time be requested by other demands, thus leading to an occasional conflicts. Suppose that N is the number of pre-assigned LSPs for the traffic flow of a faulty LSP. Our approach has nothing to do with backup LSPs selection. Selecting backup LSPs is done during connection setup time at network management interface level. In our distributed approaches, backup LSPs will be pre-established before a fault occurs. This approach has several advantages: no topology information is necessary, the condition of backup LSPs are discovered when they are needed; and no additional work is necessary in the absence of failures. Unfortunately, the flooding mechanism can involve many messages and substantial delays to discover the best LSPs.

Dynamic searching is unnecessary if a dynamic link-state routing protocol such as OSPF (open shortest path first) is being used to distribute link-state information globally. In that case, each LSR will be able to store complete network topology and status information, and therefore make a consistent selection of optimal backup LSPs. Compared to dynamic searching, the number of messages and delay can both be reduced by pre-establishing backup LSPs before it is needed, at the same time that the active LSP is established. The approach is to prepare the backup LSPs as much as possible before it is needed, so that minimum work must be done when a fault occurs (figure 1). Backup LSPs are selected and the routing tables are configured with the backup LSPs information (including label assignments). Although the backup LSPs are pre-established, band width does not necessarily have to be reserved along the backup LSPs.

Reserved backup resources will ensure that failed FECs (Forwarding Equivalence classes) can be certainly restored, but may unnecessarily restrict the admissible traffic load. For better efficiency, we will use only distributed approaches where the backup LSPs are pre-established but backup bandwidth is not reserved. Hence, there will be a possibility that the pre-selected backup LSPs will be inadequate at the time when it is needed, in which case more alternative LSPs will have to be found. The traffic of the faulty LSP is shown by a (t), and the percentage of the traffic forwarded into the i-th alternative path is represented by α_i , where we have $\sum_{i=1}^{N}\alpha_i = 1$.

So the new flow in the i-th LSP will be $\alpha_i.a(t)+b_i(t)$, where $b_i(t)$ is the ongoing traffic of the i-th LSP before the breakdown. The objective would be to solve the following problem.

Subject to:

$$\alpha_i.a + b_i < c_i \quad ,i = 1,\dots,N \qquad (1)$$

QoS is satisfied for all i (2)

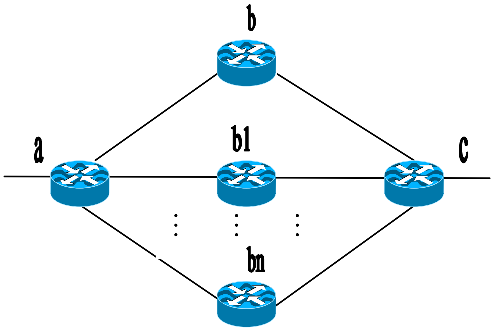

Figure 1. abc is the faulty LSP and ab_ic is the Pre-assigned Protection LSP

Where constraint (1) is the LSP capacity constraint and is for stability, while constraint (2) is the QoS constraint, where $\alpha = (\alpha_1,\dots,\alpha_N)$ is vector of all resource assignments, and h is the set of all points located on the hyper plane $\sum_{i=1}^{N}\alpha_i = 1$, and positive quadrants, namely $h = \{\alpha \mid \alpha_i > 0, i = 1,\dots,N, \sum_{i=1}^{N}\alpha_i = 1\}$ where, a and b_i are the mean rate of the broken LSP and the i-th alternative path, and c_i is the capacity of the i-th LSP. If constraint (1) is violated, the system will be unstable and the corresponding queue size will grow to infinity.

Constraint (2) is to guarantee a certain level of desirable QoS for users. If the problem has not any solution by the mentioned conditions we allow the faulty LSP to be totally disconnected and discard its traffics. Here, we use some kind of priority so the traffics with higher priority will get a better chance to be served on a link break down. The lower priority either should wait in the boundary of the network or totally be refused to access to the network.

In this case, if we have to let some traffic be thrown away, h is then to be redefined as $h = \{\alpha \mid \alpha_i > 0, i = 1,\dots,N, \sum_{i=1}^{N}\alpha_i \le 1\}$.

3.2. The Proposed Approach Based on CBR

Case-based reasoning try to make use of past experience during problem solving. CBR deploy case bases of previously solved problems to tackle new problems. When a new problem is encountered, it is used as a probe to retrieve a similar problem case from the case base. The soloution for this case is then modified or adapted to fit the target problem. CBR can also learn by storing the results of a successful problem-solving episode in its case base to be used in the future [20].

The cases are associated with qualitative and quantitative parameters, called features or attributes , and their values which is gathered in a situation as shown in figure 2.

The CBR algorithm determines the similarity between cases based on the similarity between the two cases' features. The following are common methods to calculate local similarity:

$$Numeric \rightarrow Sim(a,b) = \frac{|a-b|}{rang}$$

$$Symbolic \rightarrow Sim(a,b) = \begin{cases} 1 & if \quad a = b \\ 0 & if \quad a \neq b \end{cases}$$

$$Multi-valued \rightarrow Sim(a,b)=\frac{Card(a)\cap card(b)}{card(a\cup b)}$$

$$Taxonomy \rightarrow Sim(a,b)=\frac{h(Common\quad node(a,b))}{\min(h(a),h(b))}$$

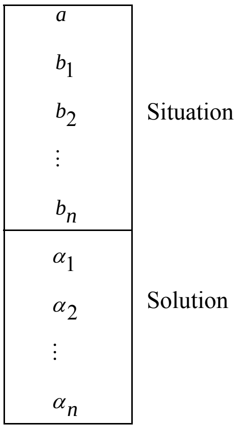

Figure 2. The case structure

Where card is the cardinality (Size) of the set, rang is the absolute value of difference between the upper and lower boundary of the set and h is the high (number of levels) of taxonomy tree. The two common similarity functions which is usually used to as numeric distance can be defined in following:

For two ρ-dimensional data objects $i=(x_{i1},x_{i2},\ldots,x_{ip})$ and $j=(x_{j1},x_{j2},\ldots,x_{jp})$, the two common similarity functions which is usually used to as numeric similarity function can be defined in following:

1- Euclidian Distance Function

$$Sim(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\ldots+|x_{ip}-x_{jp}|^2}$$

2- Manhatan Distance Function

$$Sim(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\ldots+|x_{ip}-x_{jp}|$$

When using the Euclidian distance function to compare distances, it is not necessary to calculate the square root because in our approach, the attributes of the cases are LSP's traffic rate and because the distances are always positive numbers and as such, for two distances,

$$d_1,d_2,\sqrt{d_1}>\sqrt{d_2}\Leftrightarrow d_1>d_2$$

Also because of more computational cost of Euclidian in contrast to manhatan

similarity functions, it is preferable using the second one as describes in (3). With respect to why Manhatan can work better than Euclidean, both functions were run and the following results were obtained: a: Computational time consumption for Manhatan distance function is equal to 1.474 μs , b : Computational time consumption for Euclidean distance function is equal to 2.014 μs. Therefore Manhatan distance function seems to be better choice. In our approach, the following similarity function is used to retrieve the similar cases [27,28].

$$\theta_i(\lambda_i',\lambda_i)=\frac{|\lambda_i'-\lambda_i|}{\max|\lambda_i'-\lambda_i|} \qquad (3)$$
$$0\le\theta_i(\lambda_i',\lambda_i)\le 1$$

Where λ_i',λ_i respectively stand for values of attribute in the retrieved case's situation (figure2) and the same attribute for the new problem.

Figures 2 illustrate an example of an on-line situation together with its similar cases, and the procedure essential to retrieval.

The system solves new problems as follows: first the user request is presented to the system as the input information. The system then retrieves the most similar cases from the case library, and tries to adapt the corresponding solutions such that it could best fit the ongoing situation through compositional adaptation. Since many cases at the same time can be similar to the user request, and through this way, the possibility will exist to combine the corresponding solutions in an efficient way yielding the final solution. In compositional adaptation, solutions from multiple cases are combined to produce a new composite solution [29]. In the following a framework for similarity assessment is proposed. It assumes that cases are described by sets of attribute and that the similarity of different attributes is measured independently. The similarity between two objects is measured as a weighted sum of the similarity with respect to their attributes. To be able to do that a weight and a distance measure has to be provided for each attribute. More formally:

Let, $O=\{o_1,\ldots.o_n\}$ be the set of cases whose similarity has to be assessed.

o.att: returns the value of attribute att for case o E O θ_i denotes the distance function of the i-th attribute, w_i denotes the weight for the i-th attribute. Based on these definitions, the similarity ρ between two cases o_1 and o_2 is computed as follows:

$$\rho(o_1,o_2)=\sum_{i=1}^{m}w_i\left[1-\theta_i(o_1.att_i,o_2.att_i)\right]/\sum_{i=1}^{m}w_i \qquad (4)$$

Where m is the number of attributes in a case as shown in the situation field in figure 2, $o_1.att_i,o_2.att_i$ are the ith features of the input case problem and the retrieved case respectively. w_i is the weight for the ith feature, the higher value of w_j, the more important the feature is. The higher the similarity score, the more similar two cases are. After ranking all cases in the case base by the similarity between each case and the new ongoing case, the cases that has the high similarity with the ongoing case, will be returned to reuse process. As more cases are added to the case library, the similarity computation and the resulting case ranking should become more accurate. The compositional adaptation in our approach is performed through averaging the transformed values of each attribute in the solutions of retrieved cases. The transformed value of an attribute (the i-th attribute in a solution) is determined through following expression as discussed in[23] :

$$\alpha_i'=\frac{\sum_{j=1}^{L}\alpha_i\rho_j}{\sum_{j=1}^{L}\rho_j} \qquad (5)$$

Where j and i stand for number of cases retrieved, and number of pre-assigned LSPs respectively and α_i is evaluated in the process of generating training /test set as following:

$$\alpha_i = a.\frac{\mathrm{spareT}_i}{\sum_{i=1}^{n}\mathrm{spareT}_i} \tag{6}$$

Where n=6 and spare T_i is the spare allocated capacity for the i-th pre-assigned LSP.

As it is observed in figure 2(n=7), the main abc LSP is protected by seven pre-assigned backup LSPs; $ab_1c,...,ab_7c$,where in formula (1), for abc LSP, m varies between 1000&1500 packet/second and the maximum capacity for each ab ^{i} c LSP is mentioned in Table 1.

Table 1. Pre-Assigned LSP Capacity (Racket/$_{sec}$)

LSP name	ab_1c	ab_2c	ab_3c	ab_4c	ab_5c	ab_6c	ab_7c
Capacity	500	600	700	800	900	1000	1100

All LSRs have exponential service time distribution with service rate of 3500 packet/second.

4. Experimental Results

4.1. Experimental Result Using CBR

For evaluating the CBR performance in the traffic decomposition on the basis of preserving the service quality, first of all, solve 4700 problems with 250 case number in library case. In the second step 4450 problems with 500 case number are solved and in this order, the number of problems are reduced and the number of cases in the case library are increased.

The performance is calculated according to:

$$\mathrm{performance} = \frac{\mathrm{Re\,tain\,case\,\#}}{\mathrm{Testset\,\#}} \tag{7}$$

Where the retain case # is equal to the number of problems, having successful solutions, and satisfying conditions 1, 2 and it is finally saved as suitable cases in the case library. Figure 3 shows that with 5 backup LSPs ab_ic (i=1,…,5) and 4250 samples in the case library, the performance curve reaches its saturational characteristic point. Figure 4 shows the same feature with 6 backup LSPs ab_ic (i=1,…,6) and 3500 samples in the case library.

In figure 5, we have 7 backup LSPs and 2900 samples in the case library. The performance curve reaches its saturational characteristic point. The number of cases existing in the saturational points is the number of sufficient samples in the case library using as data base in CBR for decision making of problem solving.

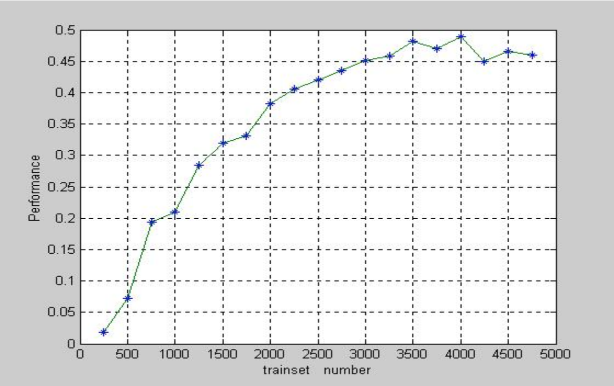

Figure 3. Performance evaluation with 5 backup LSPs

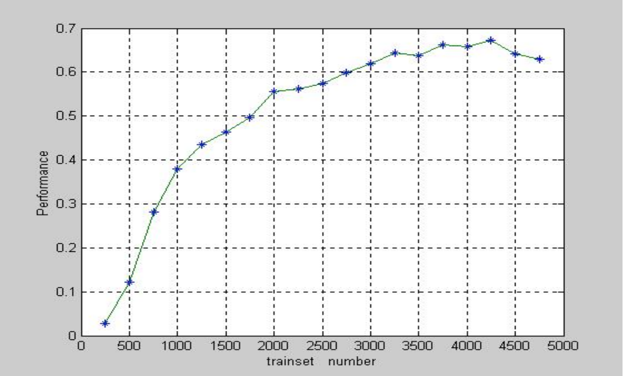

Figure 4. Performance evaluation with 6 backup LSP

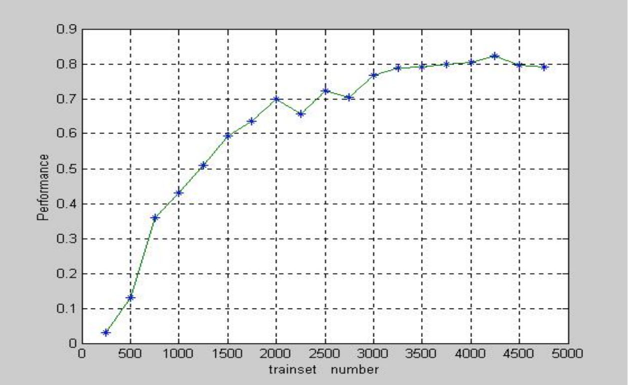

Figure 5. Performance evaluation with 7 backup LSPs

4.2. Experimental Result Using Neural Network

Several global neural network architecture have been developed over the last few decades, including the multilayer perception (MLP), the case code correlation network, and the radial basis function (RBF) network. Since the MLP has emerged as the most successful network[30], we limit our attention to MLP alone. MLPs have several significant advantages over conventional approximations.

Multilayer feed-forward networks consist of units arranged in layers with only forward connections to units in subsequent layers. The connections have weights associated with them. Each signal traveling along the link is multiplied by the connection weight. The first layer is the input layer, which contains the case situation, and the input units distribute the inputs to units in subsequent layers. In the following layer, each unit sums its inputs and adds a bias or threshold term to the sum and non linearly transforms the sum to produce an output.

This nonlinear transformation is called the activation function of the unit. The out put layer units often have linear activations[21]. The layers sandwiched between the input layer and out put layer are called hidden layers, and units in hidden layers are called hidden units. The training data set consists of N_Ω training patterns $\{(x_p, t_p)\}$, where P is the pattern number which in our approach is equal to 4000 and are executed off line in our simulation model.

The inputted vector x_p and desired out put vector t_p have dimensions N and M, respectively. As shown in figure ۲, inputted vector and output vector are situation and solution case structure parts respectively with defined values N=7 and M=6.

Y_P is the network out put vector for the path pattern. The thresholds are handled by augmenting the input vector with an element $x_p(N+1)$ and setting it equal to one. For the jth hidden unit, the net input $net_p(j)$ and the out put activation $o_p(j)$ for the path training pattern are:

$$\mathrm{net}_p(j)=\sum_{i=1}^{N+1}\omega(i,j).x_p(i) \qquad ,1 \quad \le j \le N_n \tag{8}$$

$$o_p(j)=f\left(\mathrm{net}_p(j)\ \right) \tag{9}$$

Where $w^{(i,j)}$ denotes the weight connecting the ith input unit to the jth hidden unit. For MLP network, a typical activation function f is the Sigmond $f\left(net_p(j)\ \right)=\frac{1}{1+e^{-net_p(j)}}$.

For trigonometric networks, the activations are Sines and Cosines. The K_{th} out put for the P_{th} training Pattern is Y_{PK} and is given by

$$Y_{PK}=\sum_{i=1}^{N+1} w_{io}(k,i).x_p(i)+\sum_{j=1}^{N_h} w_{ho}(k,j).o_p(j) \tag{10}$$
$$1\le K\le M$$

Where $w_{io}(K,i)$ denotes the output weight connecting the ith input unit to the Kth output unit and w_{ho} (k,j) denotes the out put weight connecting the jth hidden unit to the out put time unite. The mapping error for the P_{th} pattern is:

$$E_P=\sum_{K=1}^{M}[\ t_{PK}\ -Y_{PK}\ ]^2 \tag{11}$$

Where t_{PK} denotes the K_{th} element of the P_{th} desired out put vector. In order to train neural network in batch mode, the mapping error for the K_{th} out put unit is defined as

$$E(K)=\frac{1}{N_\Omega}\sum_{P=1}^{N_\Omega}[t_{PK}-Y_{PK}]^2. \tag{12}$$

The overall performance of MLP neural network, measured as mean square error (MSE), can be written as

$$E=\sum_{K=1}^{M} E(K)=\frac{1}{N_\Omega}\sum_{P=1}^{N_\Omega} E_P. \tag{13}$$

After training process, 950 problem as an inputted vector was tested with MLP through variable number of hidden layers. A variety of MLPs with different numbers of layers were activated where performances are shown in figure 6 . As can be seen from the figure, the performance gets better and better as the number of layers is increased until it is almost 5, where performance is at its peak. However, even under this circumstance the performance is still lower than what can be achieved using CBR.

This is because the very patterns (4000 pattern) needed to train the neural network is still relatively low due to the fact that a huge number of faults may occur in the communication network. We can therefore conclude that in such circumstance CBR can function better due to its ability to utilize compositional adaptation of solutions pertaining to similar cases.

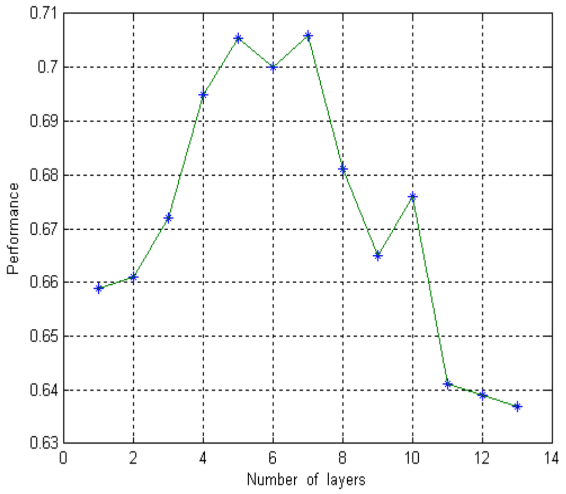

Figure 6. Performance evaluation using MLP

5. Comparison

Haskin and Makam have proposed a restoration scheme applied for MPLS network(figures 7,8) [3,17]. According to previous categorization of restoration scheme, they use a pre-assigned backup path restoration scheme. Haskin's scheme has some advantage of having a simple architecture, but it has the shortcoming of wasting network resources and crank back thus always occurs. Ingress LSR must have a large-capacity buffer to queue newly incoming traffics until the reverse LSP traffic flows are restored. There are two basic models for path recovery: rerouting and protection switching. Rerouting is a model that establishes a recovery path after a failure on its working path. Protection switching is a model that establishes a recovery path prior to any failure on the working path. According to how the repairs are affected upon the occuraence of a failure on the working path, there are two ways: global repair and local repair as shown in figure 9. In global repair, protection is always activated on end-to-end basis, irrespective of where a failure occurs. But in local repair, protection is activated by each LSR that has detected a failure. Rerouting model is very good in resource utilization. But it has the disadvantage that requires longer times or even fails in establishing a recovery path. Because of the problem, most of the existing schemes based on rerouting model considers local repair. The intent of local repair is to protect against a single link fault. So those schemes has the disadvantage of difficulty in handling node failures or concurrent node faults. This is why popular approach in MPLS fault recovery is global repair, which is used to protect against any link or node fault on entire path or on a segment of a path. In global repair,protection switching is the preferred approach because the setup overhead and delay may be greater disadvantages when longer alternative paths are sought. So almost every existing schemes based on global repair adopts protection switching. However, those schemes have some difficulty in solving problem such as resource utilization and both working and recovery path failure. Our scheme proposes an efficient rerouting mechanism to establish a LSP along the least-cost recovery paths of all possible alternative paths that can be found on a working paths. Table 2 compares the 3 restoration schemes.

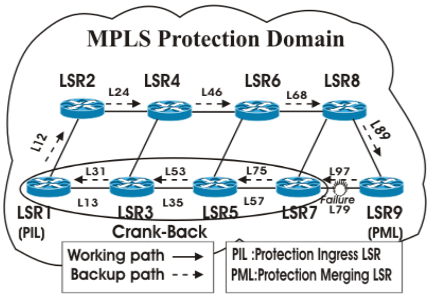

Figure 7. Haskin's scheme

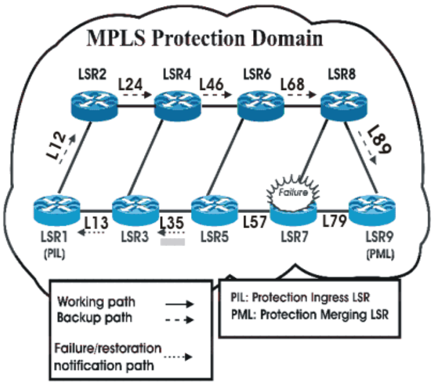

Figure 8. Makam's scheme

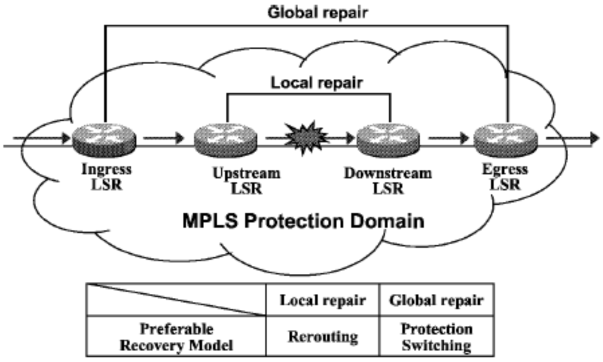

Figure 9. Global repair vs. local repair

Table 2. Performance Evaluation and Comparision of Each MPLS Recovery Scheme

Comparision	Haskin	Makam	Proposed scheme
Recovery model	Protection switching	Protection switching	Rerouting
Recovery support	End-to-end	End-to-end	End-to-end
Packet loss	Good	Bad	Good
Reordering of packets	Bad	Good	Good
Resource utilization	Bad	Normal	Good
Failure on both working and recovery path	Not support	Not support	Support
Bandwidth reservation	No	Yes	Yes
Set up alternative LSP	Yes	Yes	Yes

6. Conclusion Remarks

In mission critical MPLS networks, it is very important to quickly reroute traffic around a failure or congestion in a LSP. This paper proposed a rerouting scheme using the least cost based recovery path, which can increase resource utilization, established a recovery path relatively fast support all failures types such as link failure, node failures, failures on both a working path and its recovery path, and concurrent faults.

Disadvantage of the MLP relative to conventional approximations include its long training time and its sensitivity to initial weight values. In addition using MLP have the following problems:

1- MLP training algorithms are excessively time - consuming and do not always converge.

2- MLP training error Vs. network topology is unknown. Selecting the topology for a single hidden layer MLP is reduced to choosing the correct number of hidden units N^{n}. If the MLP does not have enough hidden units, it is not sufficiently complex to solve the function approximation problem and it underfits the data, producing excessive error at the outputs. MLP testing error is difficult to predict from training data. Determining the optimal amount of training for an MLP is difficult. A common solution to this problem involves stopping the training when the validation error starts to increase. However, this approach does not guarantee that optimal performance on a test set has been reached.

MPLS is a switching technology that presents advantages of traffic engineering and QoS support on IP networks. In this context, engineering of MPLS networks for efficient use of resource is a critical problem. A comparison was made between the performance evaluations provided by the neural networks and CBR. We show that by carefully distributing the traffic load of working LSP over network resources of back up LSPs, using CBR, performs better QoS guarantee of faulty working LSP than neural network.

References

[1] E. Rosen, A.Viswanathan, and R. Callon, "Multiprotocol label switching architecture," Network working Group, RFC 3031, 2001.

[2] R. Callon, N.Feldman, A. Fredette, G. Swallow, and A. Viswanathan, "A Framework for Multiprotocol Label Switching," http://tools.ietf.org/html/rfc4378, 1999.

[3] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. srinivasan, "RSVP-TE: Extensions to RSVP for LSP tunnels," Network working Group, RFC 3209, 2001.

[4] D. Haskin, R. Krishnan, R. Boyd, and A. Hannan, "A Method for Setting an Alternative LSPs to Handle Fast Reroute," http://tools.ietf.org/html/draft-haskin-mpls-fast-reroute-01, 1999.

[5] R. Krishnan and D. Hastin, "Extensions to RSVP to Handle Establishment of Alternate Label-Switched-Paths for Fast Reroute," http://tools.ietf.org/html/draft-krishnan-mpls-reroute-rsvpext-01, 1999.

[6] V. Sharma and F. Hell Strand, "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery," Network working Group, RFC 3469, 2003.

[7] P. Pan, D. Gan, G. Swallow, J. Vasseur D. Cooper, A. Atlas, and m. Jork, "Fast Reroute Techniques in RSVP-TE," http://tools.ietf.org/html/draft-ping-rsvp-fastreroute-00, 2003.

[8] S. Kini, M. Kodialam, S. Sengupta, and C. Villamizar, "Shared Back up lLabel Switched Path Restoration," http://tools.ietf.org/html/draft-kini-restoration-shared-backup-01, 2001.

[9] M. S. Kodialam and T. V. Lakshman, "Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels Using Aggregated Link Usage Information," *Proc. of the Annual IEEE International Conference on Computer Communications*, pp. 376-385, 2001

[10] J. Vasseur, A. Charny, and F. Le Faucheur, "Traffic Engineering Fast Reroute: by pass Tunnel Path Computation for Bandwidth Protection," http://tools.ietf.org/html/draft-vasseur-mpls-backup-computation-02, 2003.

[11] J. shi and J. P. Fonseka, "Hierarchical Self-Healing Ring," *IEEE/ACM Transaction on Networking*, vol. 3, no. 6, pp. 690-697, 1995.

[12] W. D. Grover, "Case Studies of Survivable Rings, Mesh and Mesh-Arc Hybrid Networks," *Proc. of the IEEE Conference and Exhibition on Global Telecommunications*, pp. 633-638, 1992.

[13] C. Thomassen, "On the Complexity of Finding a Minimum Cycle Cover of a Graph," *SIAM journal of computing*, vol. 26, no. 3, pp. 675-677, 1997.

[14] W. D. Grover and D. stamatelakis "Cycle-oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Reconfiguration," *Proc. of the IEEE International Conference on Communications,* pp. 537-543, 1998 .

[15] G. Ellinas and T. E. stern, "Automatic protection Switching for Link Failures in Optical Networks with Bi-Directional Links," *Proc. of the IEEE Conference and Exhibition on Global Telecommunications*, pp. 152-156, 1996.

[16] M. Merdad, S. G.Finn, and R. A. Barry, "WDM Look-back Recovery in Mesh Networks," *Proc. of the Annual IEEE International Conference on Computer Communications*, vol. 2, pp. 752-759, 1999.

[17] M. Merdrad, R. A. Barry, S. G. Finn, W. He, and S. Lumetta, "Generalized loop-back recovery in optical mesh networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 153-164, 2002.

[18] C. Chekuri, A. Gupta, A. Kumar, j. Naor, and D. Raz, "Building edge- failure resilent networks," Proc. of the Conference on Integer Programming and combinatorial optimization, pp. 439-456. 2002.

[19] R. Bhatia, M. Kodialam, and S. Sengupta, "Bandwidth Guaranteed Routing with Fast Restoration Against Link and Node Failures," *IEEE Transactions on Networking*, vol. 16, no. 6, pp. 1321-1330, 2008.

[20] B. Smyth, M. Keane, P. Cunningham, "Hierarchical Case-Based reasoning Interacting Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design," *IEEE Transaction on Knowledge and Data Engineering*, vol. 13, no. 5, pp.793-812, 2001.

[21] J. Yang, K. Shen, and C. Ong, "Feature Selection for MLP Neural Network: The Use of Random Permutation of Probabilistic Outputs," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1911-1922, 2009.

[22] K. Sohn, S. Nam, and D. Sung, "A distributed LSP Scheme to Reduce Spare Bandwidth Demand in MPLS Networks," *IEEE Transactions on Communications*,vol. 54, no. 7, pp. 1277-1288, 2006.

[23] A. Dana, A. K. Zadeh, M. E. Kalantari, and K. Badei, "Fault Recovery in MPLS Network Using Case-Based Reasoning," *Journal of Modarres Technical Engineering*, vol. 16, pp. 127-138, 2005.

[24] D. Wang and G. Li, "Efficient Distributed Bandwidth Management for MPLS Fast Reroute," *IEEE/ACM Transaction on Networking*, vol. 16, no. 2, pp. 486-495, 2008.

[25] Q. Zheng and M. Gurusamy, "LSP Partial Spatial-Protection in MPLS over WDM Optical Networks," *IEEE Transactions on Communications*, vol. 57, no. 4, pp. 1109-1118, 2009.

[26] J. Tapolcai, P. Ho, and A. Haque, "TROP: A Novel Approximate Link –State Dissemination Framework for Dynamic Survivable Routing in MPLS Networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 19, no. 3, pp. 311-322, 2008.

[27] R. Lopez de Mantaras, "A Distance-based Attribute Selection Measure for Decision Tree Induction," *Machine Learning*, vol. 6, no. 1, pp. 81-92, 1991.

[28] E. Plaza, "Case as Terms: A Feature Term Approach to the Structural Used Representation of Cases," *Proc. of the International Conference on Case-based Reasoning*, pp. 265-276, 1995.

[29] W. Wilke and R. Bergmann, "Techniques and Knowledge used for Adaptation during case-based problem solving," *Tasks and Methods in Applied Artificial Intelligence*, PP. 497-505, 1998.

[30] Z. Huang and A. Kuh, "A Combined Self-organizing Feature map and Multilayer Perceptron for Isolated Word Recognition," *IEEE Transactions on Signal Processing*, vol. 40, no. 11, PP. 2651-2657, 1992.

Arash Dana received his B.Sc and M.Sc.in Electrical Enginnering from Islamic Azad University respectively in 1993 and 1996 and Ph. D in Electrical Engineering from Islamic Azad University, Science & Research Branch, Tehran, Iran in 2003.

Dr. Dana is a lecturer at Tehran universities. His research interests are in the areas of quality of service provisioning in high speed networks, wireless communications with special emphasis on design of reliable and efficient wireless networks, ad-hoc wireless network, self-similarity and traffic modeling.

E-mail: dana.arash@gmail.com

Paper Handling Data:

Submitted: 08.26.2009
Received in revised form: 10.20.2010
Accepted: 01.23.2011
Corresponding author: Dr. Arash Dana,
Department of Electrical Engineering, Islamic Azad University-Central Tehran Branch, Tehran, Iran.