Bandwidth Management of Multiple LSP Against Link Failures Using WMMF

P. Praba[1], Dr. V. Sundaram[2], S. Veni[3]

ABSTRACT

Local Restoration has always been a main concern in MPLS networks. The main task of the Multi Protocol Label Switching (MPLS) recovery mechanism lies in controlling the traffic flow in the network and finding a backup path such that failed links or nodes can be bypassed locally from the first node that is upstream from the failures. The bandwidth reserved on each link along the backup paths are expensive and can be shared by all the service LSPs provided the protected failure points are not expected to fail simultaneously. This paper examines the traffic engineering problem of flow routing and fair bandwidth allocation where flows can be split to multiple paths. The contribution of this paper is an algorithm for per commodity weighted max-min fair rate vector. The algorithm is a fully polynomial epsilon-approximation (FPTAS) algorithm and is based on a primal-dual alternation technique. The approximation algorithm maximizes throughput when the routed traffic is required to be locally restorable.

Index Terms— Local restoration, MPLS, backup path, routing, traffic engineering.

I. INTRODUCTION

As service providers move all their applications to IP/MPLS backbone networks, bandwidth guaranteed paths with fast restoration capability becomes more and more crucial. In recent times, MPLS has gained a lot of attention due to increased restoration flexibility and high reliability for services. In MPLS [1], the ingress LSR encapsulates the packet with labels and forwards the packets along label switched paths (LSPs). These LSPs behave like virtual traffic trunks to carry the flow aggregates belonging to same "forwarding equivalence classes". Traffic engineering aims at network utilization and resource in order to achieve goals, such as, maximum flow or minimum delay and to allow different flows to share the network, so that the total flow will be maximized while preserving fairness. The flow aggregates when combined with explicit routing of bandwidth guaranteed LSPs enables service providers to traffic engineer their networks [2]. Fast restoration allows backup paths to be setup simultaneously with the active path thus ensures quick restoration of a LSP upon failure.

Two ways of restoration namely end-to-end or path restoration and local restoration are often found in the literature. In end-to-end restoration, a disjoint backup path from the active path is provided from the source to the destination for each request. When there is a link or node fails, the failure information has to propagate back to the source which in turn switches all the demands to the backup path as illustrated in Figure 1 and in Figure 2. This information propagation to the source makes it unacceptable for many applications. *Local restorability* means that upon a link or node failure, the first node upstream from the failure must be able to switch the path to an alternate preset outgoing link so that path continuity with bandwidth guarantees is restored by a strictly local decision. In the case of local restoration, the backup paths are setup locally and therefore failure information does not have to propagate back to the source before connections are switched to the backup path.

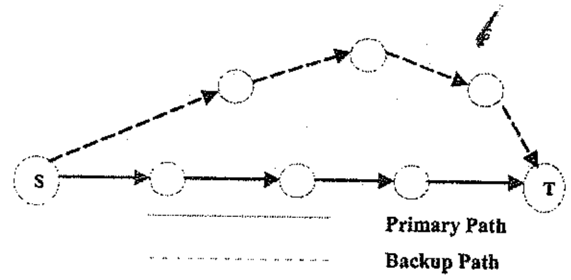

Fig 1 : Path Restoration

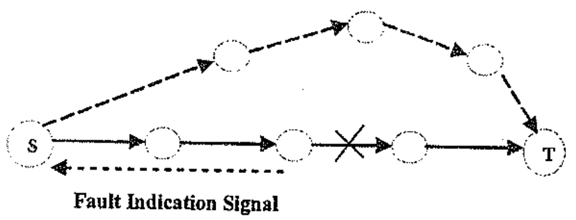

Fig 2 : Path Restoration on Link failure

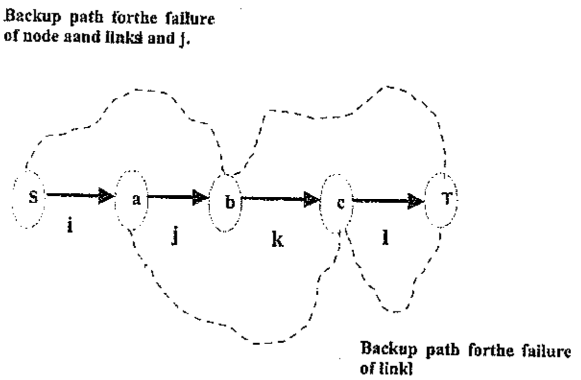

Fig3 : Backup Path for Single Element Failure

Local Restoration has to protect against single link or single element (node or link) failure. The amount of resources needed to provide backup for the second mode will be greater since a node failure results in multiple link failures. In the single link failure, when a link fails, the two nodes that are at the end point of the link detect the failures, and they immediately switch all the demands on to the backup path. In a single element failure model as shown in figure 3, when a node fails assume that all the links incident on this node fails. This is detected as in the link failure case and the demands are routed across the failed node. The important performance metrics for restoration schemes are low restoration latency, low restoration overhead in capacity usage, operational simplicity and the additional computational load imposed on the network elements in order to route restorable connections.

In this paper we consider routing with local restoration of link and node failures so as to maximize network throughput. We consider as input a network topology and directional link capacities, a list of ingress-egress pairs, and per-pair traffic demand. We assume that the given demands are aggregates of (e.g., TCP) connections, such as client traffic (university campus, business client) and will typically be expressed by average or maximum required rates. Thus, traffic between an ingress-egress pair may be split arbitrarily among different paths without causing packet reorder in the connections comprising each demand. One way to maximize the network flow is to formulate the problem as a Maximum Multi-Commodity Flow (*MMCF*) problem which can be solved using linear programming (LP).While the solution will maximize the flow, it will not always do it in a fair manner. Flows that traverse several congested links will be allocated very little bandwidth or none at all, while flows that traverse short hop distances will receive a large allocation of bandwidth. In an attempt to introduce fairness into the maximum flow problem, the Maximum Concurrent Multi-Commodity Flow (*MC* MCF) problem was suggested. However, the achieved solution under-utilizes the network, sometimes saturating only a small fraction of it. The max-min fair allocation strikes a balance between fairness and the need to fully utilize the network.

This work focuses on an extended version of the max-min fair allocation where the primary and backup flow between two terminals may be split among several paths.

The WMCM (Weighted Max-min fair Concurrent MCF) can achieve max-min fair rate vector in a polynomial number of steps and solves iteratively the maximum concurrent LP until network saturation is achieved where each iteration performs the MCMCF LP over the residual capacity of the network with the commodities whose net flow can still be increased.

The rest of the paper is organized as follows. Section II presents the background and related work. Section III defines the problem statement and describes the fast combinatorial algorithm (FPTAS) for solving the routing with restoration problem. In Section VI, we present weighted min max fair concurrent algorithm for single link failures and analyze their performance in Section V. Section VI concludes the paper and suggests future work.

II. Background Study And Related Work

Recently there has been a great deal of work addressing restoration functionality and schemes in IP/MPLS networks[3]–[6], [11]–[14], [16] as well as how to manage restoration bandwidth among network and select optimized restoration paths [7]–[10]. However, all of them deal with path-based end-to-end LSP restoration and routing protocol fast re-convergence [13], [14]. When MPLS restoration times must be comparable to SONET restoration times, the MPLS local restoration [2] is a faster alternative to path restoration. Local restoration implies that to successfully route a path set-up request, an active path and a bypass backup path for every link and node used by the active path must be determined. Since the backup LSPs are pre-established, even though they do not consume bandwidth before failure happens, the network has to reserve enough restoration bandwidth to guarantee the LSP restoration for any failure.

Many algorithms deal with local restoration of link and node failures so as to maximize network throughput. [6][8] presents dynamic routing of locally restorable bandwidth path. Pre-provisioning of bandwidth for fast restorable connection is implemented in [18], [19]. [17] suggests dynamic provisioning and restoration of light path using online algorithm for route computation. Backup path selection with the signaling extension is discussed in [15].

While there are many different algorithm known for local restoration, very few algorithm deal with maximizing the network flow [15], [17-19]. One way to maximize the network flow is to formulate the problem as a maximum concurrent multicommodity flow problem which can be iteratively solved using FPTAS [20]. However they underutilize the network. Also it saturates only a small fraction of its links as they has to satisfy the maximum equal fraction of all demands.

A min max fair allocation for multicommodity flow is presented in [21], but it does not deal with link or node failures. The contribution of this paper is a variable routing with local restoration it also focuses on optimized weighted min max fair bandwidth allocation for fast restoration connections.

III. Notations And Problem Formulation

The **Maximum Concurrent Multi-Commodity Flow** problem is stated as follows. Let $G = (N,E)$ be a directed graph with nodes N and edges E. $\forall$ e $\in$ E. We will refer to a link by e instead of (i,j) and nonnegative capacities by c(e). Traffic demands (unidirectional) between source-destination pairs in the network are given. Each of the (non-zero) traffic demands between source-destination nodes will be referred to as a commodity. There are K commodities $C_1,\ldots,C_k$, each is specified by the triplet $C_i = (s_i, t_i, dem_i)$. The pair (s_i, t_i) is the source and the sink of commodity *i*, respectively and dem_i is its rate demand. Given the network topology (nodes and links), link capacities in the network, and the demand for each commodity, the objective of *restoration routing* involves computing the maximum multiplier λ (throughput), *s.t*, for $i = 1, \ldots K$, $\lambda\ dem_i$ units of the respective commodities can be simultaneously routed, from s_i to t_i for all k along *fast restorable paths* subject to flow conservation and link capacity constraints. For routing with restoration for link failures, a path $P \in P_i$ consists of a primary (working) path, denoted by *W(P)*, and link backup (detour) paths, denoted by $B_{e(P)}$, for each link *e* on *W(P)*.

Based on the definition, a path consists of both the primary as well as link backup detours. The following linear program *MCMCF* primal is a path flow formulation that assigns the maximum commodity flow to p_i, the set of all paths (primary and backup) between s_i and t_i. Let us have a look at the link protection version. For any commodity *i*, let p_i denote the set of all paths (structures) from node s_i to node t_i. Recall that a path $p \in p_i$ consists of a primary path as well as link backup detours protecting each link on the primary. Let the variable F(P) denote the traffic routed on path *P*. The total demand routed for commodity along all paths in p_i must be $\lambda.\ dem_i$ is given in (1). (2) corresponds to the total working traffic and restoration traffic on link and failure of link *e* that must be at most *c(e)*. We now formulate the path indexed linear program for the problem of routing with restoration for link failures as

maximize λ

subject to

$$\sum_{P:P\in P_i} f(P) = \lambda.dem_i\ \forall\ i \quad (1)$$

$$\sum_i \sum_{P\in P_i, e\in W(P)} f(P) + \sum_i \sum_{P\in P_i, e\in B_f(P)} f(P) \leq c(e)$$
$$\forall\ e,f\ \in E, e \neq f \quad (2)$$

$$f(P) \geq 0\ \forall\ P \in P_i \forall\ i \quad (3)$$

The following is the LP dual to the maximum concurrent flow problem. The variable w(e,f) holds the link length which is dual to each capacity constraint (2). The variable z(i) holds the shortest path per each commodity corresponding to constraint (1). The dual linear program can be written as

Minimize

$$\sum_{e\in E} c(e) \sum_{f\in E, f\neq e} w(e,f) \quad (4)$$

subject to

$$\sum_{e\in W(P)} \sum_{f\neq e} w(e,f) +$$
$$\sum_{f\in W(P)} \sum_{e\in B_f(P)} w(e,f) f(P) \geq z(i), \forall\ P \in P_i \forall i \quad (5)$$

$$\sum_{i=1}^{k} dem_i\, z(i) \geq 1 \quad (6)$$

IV. WMMF - Weighted Max-Min Fair Algorithm

WMMF is an off-line, centralized algorithm that calculates the global max-min fair vector using an LP solver. The commodity rate vector that is calculated in *OPT WMMF* is optimal, i.e., 1-coordinate-wise competitive. The *OPT WMMF* algorithm receives as input the list of commodities, Γ; the vector of demands, *dem;* and the graph, *G*. It initializes Γ_{UNSAT}, the list of unsaturated commodities (the commodities that can still increase their bandwidth assignment), to all the commodities; and Γ_{SAT}, the list of saturated commodities to null. It proceeds in iterations. In each iteration, the algorithm lays new routes per all the commodities over *G* and increases the allocated bandwidth of the unsaturated commodities by solving a number of LPs, each is a reformulation of the *MC MCF* problem.

There are a few goals to each iteration. The first goal is to maximize λ, the equal share of all the unsaturated commodities in a fair manner and under the restriction of arc capacities. The second goal is to find a routing in *G* for all the bandwidth allocated, both in the previous iterations and the current one. At the end of the iteration, some of the commodities become saturated and are thus removed from Γ_{UNSAT} and added to Γ_{SAT} The following *OPR* LP is a reformulation of the *MC MCF* problem (equations 1-2)

maximize λ

subject to

$$\sum_i \sum_{P \epsilon P_i, e \epsilon W(P)} f(P) + \sum_i \sum_{P \epsilon P_i, e \epsilon B_f(P)} f(P) \leq c(e) \quad (7)$$

$$\forall i \epsilon \Gamma_{UNSAT}, \sum_{P:P \epsilon P_i} f(P) = \lambda . dem_i \quad (8)$$

$$\forall i \epsilon \Gamma_{SAT}, \sum_{P:P \epsilon P_i} f(P) = \lambda_{\phi i}^{sat} . dem_i \quad (9)$$

$$\forall P \in P_{i=1...K}\ f(P) \geq 0,\ \lambda \geq 0$$

Eq. 8 restricts the unsaturated commodities by λ, the equal share (weighted by the appropriate demand) that all the commodities can use. The objective of the *OPR* problem is to maximize this λ that appears only in Eq.8. The allocated bandwidth to the saturated commodities was already assigned in the previous iterations. A saturated commodity *i* that became saturated in iteration φi and was assigned a bandwidth of $\lambda_{\phi i}^{sat} . dem_i$, where $\lambda_{\phi i}^{sat}$ is the value of λ that was calculated in iteration ϕi. Eq. 9 preserves the already assigned bandwidth for the saturated commodities. Eq. 7 is the per-arc capacity constraint. By solving the *OPR* problem during iteration *i*, we find the additional possible equal share of bandwidth per each unsaturated commodity and find routing for all the commodities, unsaturated and saturated.

By solving the *OPR* problem during iteration *i*, it maximizes the equal per-commodity allocation increase and routing reassignment, such that previously allocated commodities will not steal bandwidth from the unsaturated commodities by occupying their more preferential paths.

An FPTAS approximation for the *MC MCF fast restoration* problem that uses a variable-size increment technique is developed and presented in [18], *MC MCF* equations (1) and (2) formulation as the primal problem and equations (4) and (5) as the dual problem.

The algorithm iterates on the dual variables until all the shortest paths are saturated. The advantage of sticking with the dual problem is the reflection of the fairness among the commodities, and In addition to the fairness, these variables can be used to determine the saturation of a path.

The pseudo code of the *OPT WMMF* is presented below. The algorithm starts by assigning the length of each link weights $\forall e \in E, w(e,f) = \frac{\delta}{c(e)}$ where δ is a pre-computed value chosen to achieve the desired approximation value. The algorithm alternates between primal flow variables and dual length variables to fulfill the capacity-length constraint (primal Eq. 1 and dual Eq.4).

Algorithm for Routing with Restoration for Link Failures

```
WMMF_RRLF(Γ dem, G, ε)

∀e ∈ E, w(e,f) = δ/c(e)
work(e) = 0 ∀e ∈ E
bkp(e,f) = 0 ∀e,f ∈ E, f ≠ e
while (Γ ≠ NULL) do    /* STAGE */
  stageCnt++, phaseCnt=1
  lastDL=0;
  newDL=Σ_{e∈E} c(e) Σ_{f≠e} w(e,f)
  while (newDl – lastDL <1) do  /*PHASE */
    for (i= 1 to |S|) do   /* ITER: S group of diff srcs */
      Let Γ_i group of commodities start from source i
      Compute minimum cost structure P ∈ P_C, C ∈ Γ_i
                    under weights w(e,f) as described;
```

```
∀C ∈ Γ_i , tmpdem(C)= dem_C
while newDL – lastDL < 1 and ∃C ∈ Γ_i ,
            tmpdem(C)>0 do   /* STEP */
  for C_k ∈ Γ_i   /* Connectivity ans Saturation test */
   If ∀f ∈ W(P_Ck) , w(f,e)≥ 1/c(f) and
         ∀e ∈ B_f(P_Ck) , w(e,f)≥ 1/c(e) then
    Γ_i = Γ_i \ {C_k}
    Γ = Γ \ {C_k}
   else
    b = min_{f∈W(P_Ck)} (c(f), min_{e∈B_f(P_Ck)} c(e))
    Δ= min (tmpdem(C_k), b)
    f(P_Ck)= f(P_Ck)+ Δ
    tmpdem(C_k) = tmpdem(C_k) − Δ
    for each f ∈ W(P_Ck) do
     work(f)=work(f) +Δ
     bkp(e,f)=bkp(e,f)+ Δ ∀e ∈ B_f(P_Ck)
     w(e,f)=w(e,f)(1 + εΔ/c(e)) ∀e ∈ B_f(P_Ck)
     w(f,e)=w(f,e)(1 + εΔ/c(f)) ∀e ∈ E, e ≠ f
   end for
  end for
  newDL=Σ_{e∈E} c(e) Σ_{f≠e} w(e,f)
 end while
end for
phaseCnt++
end while
lastDL=newDL
end while
```

$\forall k = 1 \dots. K, \forall P \in P_k, f(P) = \frac{f(P)}{\log_{1+\epsilon} \frac{1+K\epsilon}{\delta}}$

$\forall k = 1 \dots. K, \forall P \in P_k, f(P_k) = \sum f(P)$

Returns per commodity k; Set of paths P_k and flows $f(P_k)$

It proceeds in stages. In each stage the algorithm solves the *MC MCF* problem (and finds the primal-dual (λ and D(l))solution. Part of the commodities become saturated during each stage and should be omitted in the following stages. The saturation test is an important contribution of our algorithm that promises the reduction in the number of the participating commodities at each stage and thus the convergence of the algorithm.

The stage proceeds in phases (line 8). Each phase is composed of $|S|$ iterations, where S is the group of all the sources (some commodities can have the same source s_i). Iteration i of phase j considers the commodities C_q, q = 1...r starting from the same source s_i and routes *dem* (C_q) units in a number of steps. Each step calculates the shortest path tree starting from the source, using the last calculated length variables $w(e,f)$. It iterates over the q commodities and in each step either saturates the current shortest path per commodity C_q or allocates the remained *dem* (C_q).

After the flow of value Δ is sent along path P_{ck}, the weights $w(e,f)$ and the flow value on each associated link are updated as follows:

1. For each $f \in W(P_{C_k})$, update the weights $w(e,f)\ \forall f \in E, f \neq e$, associated with this link on the primary path as

$$w(f,e) = w(f,e)\left(1 + \frac{\epsilon\Delta}{c(f)}\right)$$

2. For each $f \in W(P_{C_k})$, update the weights $w(e,f)\ \forall e \in B_f(P_{C_k})$ associated with links on the backup detour for (primary) link as

$$w(e,f) = w(e,f)\left(1 + \frac{\epsilon\Delta}{c(e)}\right)$$

3. Increment the working traffic for each link $f \in W(P_{C_k})$ and increment the restoration traffic on each backup link $e \in B_f(P_{C_k})$ that appears on failure of primary link f by Δ.

The entire stage ends as soon as D(l) ≥1 according to the dual constraint Eq. 5. The corresponding per commodity net flow vector $f(P_k)$, $k = 1....K$ is infeasible for the primal LP, and needs to be scaled down. For this purpose, we note that as long as D(l) < 1, the length of each link cannot exceed 1/c(e), which implies that the number of times the flow is increased over this link is $log_{1+\epsilon} \frac{1+K\epsilon}{\delta}$ times its real flow. By scaling down this flow by a factor of $log_{1+\epsilon} \frac{1+K\epsilon}{\delta}$ feasible flow will be achieved. The scaling is done after the termination of all the phases.

Iterating over $|S|$ is more efficient than iterating over the commodities since the entire shortest path tree is calculated at once instead of one shortest path calculation at a time. The primal-dual solutions are found when the function D(l) is larger than 1. Since, each stage is an activation of the primal-dual alternation. In order to saturate the network, we continue to increase the weight $w(e,f)$, but the termination condition (D (l) >1) should consider only the additional length for the last stage. Thus, the variables hold the accumulative length values and are used for the shortest path calculations. However, for the stage termination condition we consider only the incremental values, namely, *newDL - lastDL*, where lastDL is the D(l) value at the beginning of the stage and newDL is the current value of D(l). At each stage, at least one commodity is saturated and removed from the list since, at least, one link value is increased by a factor of $(1+\epsilon)/c(e)$. This ensures the algorithm convergence.

V. Algorithm Implementation

The algorithm was implemented using MATLAB. The capacity performance of local restoration for link failures using WWMF was analyzed. To illustrate it, we provide the simple example of figure 4.

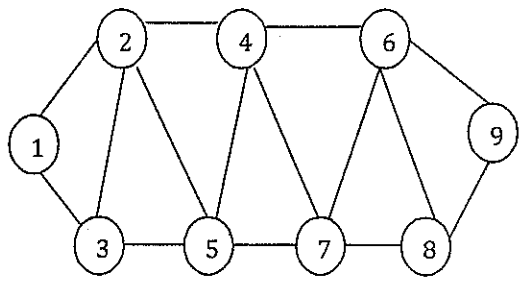

Fig. 4 : Algorithm Iteration Example

The capacity of each link is 1. There are four commodities, (1, 2), (1, 3), (2, 6) and (2, 9) each with 1 unit of demand. All the links and paths are unidirectional. Each commodity chooses the shortest path as primary path and the backup paths are setup locally to protect each link in the primary path against failures.

TABLE I
WMMF EXECUTION USING $\epsilon = 0.1$

comm. ID	min max flow rate	per comm. flow	path length
stage 1			
1	0.3297	0.3297	0.0013
2	0.3297	0.3297	0.6721
3	0.3297	0.3297	0.9532
4	0.3297	0.3297	0.7839
$\lambda = 0.3297$			lastDL=1.126
stage 2			
1	0.6589	0.6589	0.4581
2	0.3297	0.3297	∞
3	0.3297	0.3297	∞
4	0.3297	0.3297	∞
$\lambda = 0.3297$			lastDL=2.211

Table I presents the two stages of the algorithm operation for $\epsilon = 0.1$. We can see that in the first stage all the commodities receive an equal portion of their demands along primary and backup paths. Link (2, 4) along primary path and (2, 5) and (5, 4) that protects (2,4) are the bottleneck links and their length after this stage becomes 1.126. It means that this link is saturated. We can verify it by observing its flow which is 0.3297+0.3297+0.3297 = 0.9891. The calculated λ for this stage is 0.3297 and the stage terminates when D(l) = 1.126. Path 2 of commodity 2, 3 and 4 does not get any flow due to the saturation of link (2, 4). At the second stage the algorithm discovers that commodities 2, 3, and 4 are saturated and delete them from Γ. In the following stage, the algorithm iterates for commodity 1 along shortest path 1 until it saturates. The final max-min vector rate for $\in = 0.1$ and commodities 1, 2, 3 and 4 is {1.6589, 0.3297, 0.3297, 0.3297} i.e., (2/3, 1/3, 1/3, 1/3).

VI. Conclusion

We presented the problem of fast local restoration that requires efficient bandwidth management to setup bypass path for every node or link traversed by the service path. Local restoration is quick since the detours are pre-computed and failure detection is localized. In this paper, we developed an algorithm that finds the weighted max-min fair rate vector by using an LP formulation and solver. The fully polynomial epsilon-approximation (FPTAS) algorithm guarantees performance for capacity allocation and routing in MPLS with fast restoration constraints when the network traffic demands are known.

References

[1] Rosen et al., *"Multiprotocol Label Switching Architecture"*, IETFRFC 3031, 2001.

[2] Pan .P, G. Swallow, and A. Atlas, *"Fast Reroute Extensions to RSVP-TE for LSP Tunnels"*, IETF RFC 4090, 2005.

[3] Kiniet .S et al., *"Shared Backup Label Switched Path Restoration"*, IETF Internet draft, work in progress, May 2001.

[4] Sharma .V and F. Hellstrand, *"Framework for Multi-Protocol Label Switching MPLS-Based Recovery"*, IETF RFC 3469, 2003.

[5] Gruber .C, *"Bandwidth requirements of MPLS protection mechanisms"*, 7th INFORMS TELECOM, Boca Raton, FL, 2004.

[6] Kodialam .M and T. Lakshman, *"Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information"*, IEEE INFOCOM, 2001.

[7] Li, D. Wang, C. Kalmanek, and R. Doverpike, *"Efficient distributed path selection for shared restoration connections"*, IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 761–771, October 2003.

[8] Kodaliam .M and T. Lakshman, *"Dynamic routing of bandwidth guaranteed tunnels with restoration"*, IEEE INFOCOM 2000.

[9] Liu .Y, D. Tipper, and P. Siripongwutikorn, *"Approximating optimal space capacity allocation by successive survivable routing"*, IEEEINFOCOM 2001.

[10] Qiao .C and D. Xu, *"Distributed Partial Information Management (DPIM) for survivable networks"*, INFOCOM 2002.

[11] Doverspike .R and J. Yates, *"Challenges for MPLS in optical network restoration"*, IEEE Communication Magazine, February 2001.

[12] Y. Xiong and L. Mason, *"Restoration strategies and spare capacity requirementsin self-healing ATM networks"*, IEEE/ACM Transactions on Networking, vol. 7, no. 1, February 1999.

[13] Basuand J. Riecke, *"Stability issues in OSPF routing"*, ACMSIGCOMM 2001.

[14] Alaettinoglu, V. Jacobson, and H. Yu, *"Towards Millisecond IGP Convergence"*, IETF Internet draft, draft-alaettinoglu-isis-convergence-00.txt,November 2000.

[15] Dongmei Wang and Guangzhi Li, *"Efficient Distributed Bandwidth Management for MPLS Fast Reroute"*, IEEE/ACM Transactions on Networking, vol. 16, no. 2, April 2008.

[16] Sabella .R, M. Settembre, G. Oriolo, F. Razza, F. Ferlito, andG. Conte, *"A multilayer solution for path provisioning in newgeneration optical/MPLS networks"*, J. Lightw. Technol., vol. 21, no.5, pp. 1141–1155.

[17] Mansoor Alicherry, Randeep Bhatia and Yung-Chun (Justin) Wan *"Designing networks with hop bounded protection paths"*, IEEE 19th International Workshop on Quality of Service (IWQoS), 2011

[18] Randeep S. Bhatia, Murali Kodialam, T. V. Lakshman, and Sudipta Sengupta. *"Bandwidth Guaranteed Routing With Fast Restoration Against Link and Node Failures"*, IEEE/ACM TRANSACTIONS ON NETWORKING, vol. 16, No. 6, December 2008.

[19] Mansoor Alicherry and Randeep Bhatia. *"Simple Pre-Provisioning Scheme to Enable Fast Restoration"*, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 15, NO. 3, JUNE 2007

[20] Garg .N and J. Konemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in 39th Annu. Symp. Foundations of Computer Science (FOCS'98), Nov. 1998.

[21] Miriam Allalouf and Yuval Shavitt. *"Centralized and Distributed Algorithms for Routing and Weighted Max-Min Fair Bandwidth Allocation"* IEEE/ACM TRANSACTIONS ON NETWORKING, vol. 16, No. 5, October 2008.

Author's Biography

P.Praba : I have presented 15 papers in National conferences and 2 in international conferences. I have completed M.Sc., MPhil in Computer Science and pursuing doctorate degree in Computer Applications at Anna University Coimbatore under the guidance of Dr. V. Sundaram, Director, Department of MCA, Karpagam College of Engineering. My area of research includes network architecture and network protocols. I have twelve years of experience in teaching and currently working as Asst. Professor and Head in the department of Computer Science, Karpagam University, Coimbatore.

Dr. V. Sundaram received his professional degree M.Sc. in Mathematics from the University of Madras in the year 1967 and he received his Professional Doctoral Degree Ph. D in Mathematics from the University of Madras in 1989. He is currently working as Director , Department of Computer Applications in Karpagam College of Engineering, Tamilnadu, India. He is a research Guide for Anna university, Bharathiar university as well as Karpagam University in the field of Computer applications. He has published several papers in International Journals and Conferences and also published 13 books in the area of engineering mathematics and he is the life member of ISTE and ISIAM. His research interests are in Cryptography and network security, Applied Mathematics, Discrete Mathematics, Network etc.

S.Veni : I have presented 12 papers in National conferences and 2 in international conferences. I have completed M.Sc., MPhil in Computer Science and currently working as a Asst

Professor in Karpagam University. I have nine years of experience in teaching. I am Pursuing Doctorate degree in Computer Science under the guidance of Dr.G.M.Kadhar Nawaz, who is working as Director in Department of Computer Applications in Sona College of Technology, Salem. My area of research is Traffic Engineering in Multiprotocol Label Switching environment.