Fault Restoration Techniques for MPLS with QoS Constraints

Tae (Tom) Oh
Thomas M. Chen
Dept. of Electrical Engineering
SMU
PO Box 750338
Dallas, TX 75275-0338 USA
Email: taehwan@seas.smu.edu, tchen@seas.smu.edu
Tel: (214) 768-8541
Fax: (214) 768-3573

Jeffery L. Kennington
Dept. of Computer Science Engineering
SMU
PO Box 750122
Dallas, TX 75275-0122 USA
Email: jlk@seas.smu.edu
Tel: (214) 768-3088

Abstract – This paper investigates distributed fault restoration protocols for multiprotocol label switching (MPLS) that automatically reroute label switched paths in the event of failures while maintaining quality of service (QoS) requirements. Protocols for link restoration, partial path restoration, and path restoration are evaluated. A new adaptive approach to selecting backup routes is proposed to maximize the probability of successful restoration and minimize restoration time. A route selection algorithm based on optimization of equivalent bandwidth is presented and demonstrated for an example network.

Keywords: multiprotocol label switching (MPLS), quality of service (QoS), faults

January 31, 2000

1. INTRODUCTION

The IETF is establishing common agreements on a framework for multiprotocol label switching (MPLS) to integrate various implementations of IP switching that use a "label swapping" technique similar to ATM to speed up IP packet forwarding without changes to existing IP routing protocols [1-7]. A key concept in MPLS is the separation of an IP router's functions into two parts: forwarding and control. The forwarding part responsible for how data packets are relayed between IP routers is based on the concept of forwarding equivalence classes (FECs) identified by label prefixes similar to ATM virtual path/virtual channel identifiers. Upon entry into an MPLS domain, all packets are mapped into FECs based on their IP source address, destination address, or other IP header fields. A label is prefixed to the packet and removed upon egress from the MPLS domain. A label is a short, fixed-length number that is independent of the network layer (e.g., a label does not include any network layer addresses) and based strictly on mutual agreement between two neighboring MPLS-enabled routers. The label can use an existing layer 2 header field (e.g., the VPI/VCI field in the ATM cell header) or be inserted between the layer 2 and IP headers as a small "shim" label consisting of a 20-bit label value; 3-bit class of service; 1-bit bottom of stack indication; and 8-bit time-to-live (TTL) for preventing accidental looping [4].

The label-swapping technique essentially involves a table look-up of a packet's label to determine its route (outgoing interface), new label value, and additional packet-handling information (if any). Label swapping is considerably simpler than the normal datagram processing involving longest prefix matching, and thus improves the price/performance and scalability of routers. A router capable of MPLS is a label switching router (LSR), and a set of LSRs traversed by a packet is a label switched path (LSP). A contiguous set of LSRs under a single administration constitutes an MPLS domain. A packet is forwarded across an MPLS domain based only on its label with no need to re-examine the network layer packet header.

Although better scalability and faster packet forwarding are the most apparent motivations for MPLS, the approach is also attractive for the support of traffic engineering and quality of service (QoS). The fact that packets are mapped to FECs with specific routes implies that traffic can be controlled better compared to traditional connectionless IP routing [8-10]. For example, routes for FECs may be selected to distribute traffic more uniformly through the network. In addition, QoS guarantees can be supported when MPLS is used in conjunction with constraint-based routing and resource reservation. QoS assurances are important for guaranteed services with hard bound requirements on packet loss, packet delay, and delay variation (jitter) [11]. Guaranteed services also need assurances about reliability in terms of continuity of services maintained by fast rerouting around network failures [12]. The currrent Internet has a degree of survivability because dynamic routing protocols will react to faults detected from routing information updates and compute alternate routes. In contrast, MPLS is potentially more vulnerable to faults as a connection-oriented technique.

This paper investigates distributed fault restoration protocols for MPLS that quickly reroute FECs in the event of failures while maintaining QoS requirements. For fault restoration, we assume that a set of QoS mechanisms are already in place, e.g., QoS or constraint-based routing and a signaling protocol for reserving resources. Constraint-based routing is an extension of traditional shortest path routing to compute routes that satisfy a set of QoS requirements and administrative policies [13,14]. Assuming the use of extended OSPF (open shortest path first) for example, each router advertises link information including maximum link bandwidth, available bandwidth, current bandwidth reservations, administrative attributes, and a default traffic engineering metric [9]. Each router collects routing information and applies a path selection algorithm constrained to feasible paths that satisfy the requirements and policies. Two signaling protocols are being considered for MPLS: constraint-based routed label distribution protocol (CR-LDP) and extended resource reservation protocol (extended-RSVP) [15-18]. CR-LDP is assumed as the basis for fault restoration messages in this paper.

As a so-called layer 2.5 protocol, MPLS can operate over many data link layers and physical layers which may well have their own fault protection mechanisms such as self-healing in the ATM layer or automatic protection switching in the SONET/SDH layer [19]. In this case, fault restoration at the lower layers may be faster than the MPLS layer, and a coordination function must exist in the routers to avoid interference between restoration actions in the different layers. Although fault restoration capabilities in the MPLS layer may be redundant, the MPLS layer provides flexibility and ensures service reliability regardless of the lower layers which may or may not be reliable.

Different strategies for fast rerouting are described and evaluated in Section 2. In Section 3, an adaptive approach for selection of backup routes is proposed along with a route optimization algorithm. Finally, in Section 4, simulation results for an example network are presented and discussed.

2. FAULT RESTORATION TECHNIQUES

Traditional fault restoration depends on a centralized network manager to detect faults and manually reroute traffic around effected areas. This process is relatively slow, not scalable to large networks, and vulnerable to failure. Fault restoration functions can be distributed among the LSRs to automatically detect and carry out rerouting without the need for manual intervention, in a manner similar to self-healing in ATM [20-22]. Distributed fault recovery can be faster and more scalable than centralized fault recovery, at the expense of more complexity in the LSRs. LSRs must be automated with protocols to carry out these steps: fault detection, backup route selection, fast rerouting, and re-optimization after fault restoration.

Distributed approaches may differ fundamentally according to the way backup routes are selected: either found dynamically at the time of a fault or pre-established before a fault occurs. Dynamic searching usually involves flooding messages after fault detection to discover all possible backup routes, and a selection of one of them according to some criterion [23,24]. This approach has several advantages: no topology information is necessary; the condition of backup routes 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 route.

Dynamic searching is actually unnecessary if a dynamic link-state routing protocol such as OSPF 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 routes. Compared to dynamic searching, the number of messages and delay can both be reduced by pre-establishing a backup route before it is needed, for example, at the same time that the active path is established. The approach is to prepare the backup route as much as possible before it is needed, so that minimum work must be done to carry out fault restoration. At least the backup route is selected and the routing tables are configured with the backup route information (including label assignments). Only a single message is then needed to activate the backup route. Although the backup route is pre-established, bandwidth does not necessarily have to be reserved along the backup route.

Reserved backup resources will ensure that failed FECs can be certainly restored, but may unnecessarily restrict the admissible traffic load. For efficiency, we will study only distributed approaches where the backup route is pre-established but backup bandwidth is not reserved. Hence, there will be a possibility that the pre-selected backup path will be inadequate at the time when it is needed, in which case alternative paths will have to be found.

An important factor unique to MPLS is the granularity of FECs which is a design choice for the network provider. A link or node failure will effect multiple FECs which must be rerouted all at the same time. Coarse-grained FECs simplify fault restoration by allowing large bundles of traffic to be rerouted as a single group. On the other hand, fine-grained FECs allow more control over the effected traffic. For example, the rerouted traffic can be distributed more uniformly over backup routes, but there will be more overhead cost to reroute a larger number of FECs. The operating point for this trade-off in the granularity of FECs is a decision for the network provider.

2.1. LINK RESTORATION

For the fastest recovery around single link failures, the link restoration approach finds an alternate path between the two LSRs directly on the upstream and downstream ends of the failed link, as shown in Figure 1. Although one rerouted FEC is shown, all FECs from the failed link must be rerouted at the same time and may be distributed to different backup routes for traffic balance. For restoration speed and efficiency, the backup route is pre-established at the same time as the active route, but bandwidth along the backup route is not reserved in order to make it available for active traffic. The backup route is selected according to the path selection algorithm described later, and the routing tables at the LSRs along the backup route are configured with the appropriate routing table entries. Hence, when traffic is rerouted to the backup route, only changes in bandwidth allocations but not routing table changes are needed at the LSRs. Since bandwidth is not reserved, it is possible that the pre-selected backup route may have inadequate available resources when the fault occurs. To maximize the likelihood of successful restoration, we will propose an adaptive method later to regularly update the backup route selection.

The restoration process for each effected FEC consists of the following steps:

Step 1: Restoration request: After the LSR on the downstream end of the failed link detects the fault, it must check the pre-selected backup route for adequate resources by sending a Restoration Request (RR) message in the backward direction along the backup route. The RR message is derived from the Label Mapping message, a label assignment message used in constraint-based routing, with additional fields to identify the effected FEC, labels, requested QoS (packet loss ratio, packet delay bound), traffic characteristics, failed link, and downstream/upstream LSRs.

Each LSR calculates the requested equivalent capacity based on its buffer resources, requested QoS, and specified traffic characteristics. Equivalent capacity is an approach to analytically derive the precise needed bandwidth to support a specific QoS for a given traffic flow [25]. If the equivalent capacity is available, the LSR makes the resource allocation change and propagates the RR message backward to the next LSR in the backup route. If insufficient resources are available, the LSR will terminate the RR message and return a Release message in the forward direction to the LSR on the downstream end of the failed link. The Release message contains the information from the RR message, as well as the identity of the rejecting LSR and reason for rejection. The Release message will clear the resources allocated already along the backup route. When the downstream LSR receives the Release message, it will proceed to Step 2.

Step 2: Restoration reattempt: If the LSR on the downstream side of the failed link receives a Release message, it will reattempt to check for adequate resources on another candidate backup route. Assuming the use of a link-state routing protocol such as extended OSPF, the LSR has complete information about network topology and status, and there is no need for a dynamic search by flooding. The LSR will exclude the pre-selected backup path from its set of feasible alternative paths, and perform the path selection algorithm on the other feasible alternative paths to select another backup route.   Step 1 is reattempted on the second alternative path. Because this second alternative path has not been pre-established, the RR message is now source routed with the alternative path specified by the originating LSR. Source routing will relieve the LSRs from having to make routing decisions for the RR message. Also, the RR message will have to carry out label assignments, routing table updates, and bandwidth allocation changes at each LSR.

If the RR message is successful along the candidate route, the restoration process proceeds to Step 3. If the RR message is rejected by any LSR, a Release message is returned as before and Step 2 is reattempted on the remaining feasible alternative paths. When the set of feasible alternative paths are exhausted without success, the failed FEC cannot be restored with its QoS guarantees. If reliable services are critical, the network provider must ensure that sufficient backup resources will be properly provisioned in the network such that an adequate backup route will always be found even under high traffic load.

Step 3: Reroute to backup path: If the RR message is successfully propagated to the LSR on the upstream end of the failed link, it will change its routing table to reroute traffic from the failed link to the backup route.

For this entire restoration process, we are interested in the degree of restorability of a failed FEC measured by the probability that an FEC will be restored within *M* attempts, P_M, assuming that there are only *M* feasible backup paths. Restorability may be expected to increase with larger *M*, which is a design parameter for the network provider. Another performance metric of interest is the restoration time measured in the mean time to complete the three restoration steps, $E(T_r)$, and mean number of attempts, *E(m)*.

We assume that an RR message has a random probability of rejection *q* by an LSR, and acceptance or rejection of an RR message is independent between consecutive LSRs although it may be correlated in actuality. Also, the probability will vary between the first attempt and consecutive attempts; the probability of success should presumably be higher on the first attempt if the path selection algorithm is correct, but the analysis here will make the simplifying assumption that each the probability *q* is the same on each attempt. The probability of rejection *q* will be a complicated function of many factors such as active traffic load, available bandwidth and buffers, requested QoS, and traffic characteristics. Additionally, we assume that the *M* feasible backup paths will all be *N* hops where *N* is a design parameter chosen by the network provider. Normally, *N* should be kept small, say within the approximate range of 2 to 5, in order to increase the probability of restoration and minimize the number of hops for RR messages.

An important metric of performance is the probability of successful restoration on the first attempt:

$$P_1 = (1-q)^N \tag{1}$$

Under the assumptions, each attempt has the same probability of success, and the probability of eventual restoration within *M* attempts is clearly

$$P_M = 1 - (1 - P_1)^M \tag{2}$$

If backup capacity has been properly provisioned, we may expect that the probability *q* will be very small. If *q* is much smaller than *1/N*, then $P_1 \cong 1 - Nq$ and the probability of eventual restoration is approximately $P_M \cong 1 - (Nq)^M$. Naturally, it can be seen that the probability of eventual restoration can be increased to arbitrarily close to 1 by increasing the number of backup paths.

The restoration time T_r for a successful attempt is the sum of store-and-forwarding delays and processing delays at each LSR (propagation delays are ignored for simplicity but can be added easily). RR messages may be expected to be assigned high priority, so queueing delays are assumed to be minimal compared to a fixed time t_p representing packet processing (including acceptance/rejection decisions) and packet forwarding. Hence, a successful attempt will involve a total delay of Nt_p.

On any unsuccessful attempt, the RR message will be rejected at a random LSR and additional delays will be incurred by the return of the Release message. The probability of being rejected at the *i*th LSR along the backup route is $q(1-q)^{i-1}$. If the time to store, process, and forward Release messages at each LSR is a fixed time t_r and the RR message is rejected at the *i*th LSR, then the total delay involved in an unsuccessful attempt will be $i(t_r + t_p)$ including the time to forward the RR message and return the Release message. The mean total delay involved in each unsuccessful attempt is

$$E(T_{rel}) = \sum_{i=1}^{N} i(t_r + t_p)q(1-q)^{i-1} = (t_r + t_p)\left[1 - (1-q)^N (1 + Nq)\right]/q \tag{3}$$

The probability of exactly *m* unsuccessful attempts before a successful restoration is $P_1(1-P_1)^m$, $0 \le m < M$, and the mean number of unsuccessful attempts is therefore

$$E(m) = \sum_{m=1}^{M-1} mP_1(1-P_1)^m = \left[1-(1-P_1)^{M-1}(1+(M-1)P_1)\right]/P_1 \qquad (4)$$

Combining (3) and (4), the total mean restoration time including failed attempts is

$$E(T_r) = Nt_p + E(m)E(T_{rel}) \qquad (5)$$

If *q* is much smaller than *1/N*, then

$$E(T_{rel}) \cong (1-(1-N^2q^2))/q \cong N^2q \qquad (6)$$

and

$$E(m) \cong \frac{1-M(Nq)^{M-1}}{1-Nq} \qquad (7)$$

Combining (6) and (7), the total restoration time is approximately

$$E(T_r) \cong Nt_p + N^2q\frac{1-M(Nq)^{M-1}}{1-Nq} \qquad (8)$$

2.2. PARTIAL PATH RESTORATION

Although link restoration is simple, the obvious limitation of the technique is the inability to handle node failures. The downstream LSR detecting the failure will attempt to reroute to an alternative path to the neighboring LSR on the upstream end, under the assumption of a single link failure. However, that LSR might be the point of failure which would cause the link restoration process to fail. To restore around any possible upstream link and node failures, partial path restoration attempts to find an alternative route from the detecting downstream LSR to the ingress router at the edge of the MPLS domain, as shown in Figure 2.

This process may be expected to be slower than link restoration because an RR message must reach the ingress LSR. It is also more complicated because the effected FECs may have different ingress LSRs. At least a separate RR message must be sent for each ingress LSR involved.

The analysis of restorability and mean restoration time is similar to the previous section except that the value of *N* will depend on the location of the fault. If the fault occurs near the ingress LSR, the backup route may be short and restoration may be fast. If the fault occurs near the egress LSR, the restoration may be slow. For analysis, we assume that a typical LSP consists of *H* hops and a fault is equally likely with probability *1/H* to be located at any hop. In the Internet, LSPs could be quite lengthy and *H* could be much more than the value in *N* in link restoration. If the fault occurs on the *n*th hop, feasible backup routes will all consist of *n* hops. Hence the length of backup routes may range between 1 and *H* with uniform probability. The probability of successful restoration on the first attempt is now

$$P_1 = \sum_{h=1}^{H} \tfrac{1}{H}(1-q)^h = \frac{1-q-(1-q)^{H+1}}{Hq} \tag{9}$$

Again, if *M* attempts are possible, restorability measured in terms of the probability of eventual restoration is given by (2) where P_1 is now (9).

Because the length of backup routes is uniformly distributed between 1 and *H*, the mean delay in a successful restoration attempt will be $t_p \sum_{h=1}^{H} h/H$, which is approximately $t_p H/2$ for large *H*. On any unsuccessful attempt, if the backup route is *h* hops, the conditional mean delay is

$$E(T_{rel} \mid h) = (t_r + t_p)[1-(1-q)^h(1+hq)]/q \tag{10}$$

Unconditioning on the probability of *h* hops, the mean delay involved in each unsucccessful attempt is

$$E(T_{rel}) = \sum_{h=1}^{H} \frac{1}{H} E(T_{rel} \mid h) = \frac{t_r + t_p}{Hq^2}[Hq(1+(1-q)^{H+1}) - 2(1-q)(1-(1-q)^H)] \tag{11}$$

The mean number of unsuccessful attempts *E(m)* is given by (4) where P_1 is now (9). Combining (4) and (11), the total mean restoration time including failed attempts is

$$E(T_r) = t_p \sum_{h=1}^{H} h / H + E(m)E(T_{rel}) \tag{12}$$

where $E(T_{rel})$ is given by (11).

2.3. PATH RESTORATION

In partial path restoration, the segment of the original active LSP downstream from the failure is uneffected by the rerouting. For more flexibility, it might be desired to reroute the entire failed LSP to another path between the ingress and egress LSRs, as shown in Figure 3. In the path restoration approach, when an LSP fails, the LSR detecting the fault will send a Fault Notification message downstream to the egress LSR. A backup route is pre-established between the ingress LSR and egress LSR. When the egress LSR receives the Fault Notification message, it will send a RR message in the backward direction along the backup route to check for bandwidth and reserve resources. If the RR message reaches the ingress LSR successfully, the LSR will switch the indicated FEC to the backup path. If any LSR along the backup route rejects the RR message, a Release message will be sent back to the egress LSR which will try another feasible backup path.

Any ingress-to-egress alternative path may be chosen, so path restoration has more flexibility than link restoration or partial path restoration. However, the restoration time may be significantly more because the Fault Notification message must be sent to the egress LSR. If the fault occurs near the ingress LSR, the Fault Notification message would have to traverse most of the LSP. If the fault occurs near the egress LSR, then the transit delay for the Fault Notification message may not be substantial.

As before, we assume that a typical LSP consists of *H* hops and a fault is equally likely with probability *1/H* to be located at any hop. If the fault occurs on the *n*th hop, the Fault Notification message will travel *H-n* hops to the egress LSR. The transit delay for a Fault Notification message is assumed to be linearly proportional to the number of hops, so a message traveling i hops will experience it_d delay where t_d is a delay constant depending on the characteristics of LSPs. The mean delay experienced by Fault Notification messages will be $t_d \sum_{h=1}^{H} h / H$. This delay must be added with times for unsuccessful restoration attempts and a successful attempt.

First, we will need to find the restorability in terms of the probability of eventual restoration. We assume that backup routes will all have *H* hops, and there are *M* feasible backup paths. The probability of successful restoration on the first attempt is clearly

$$P_1 = (1-q)^H \tag{13}$$

Although similar in form to (1), the value of *H* in (13) may be expected to be much larger than *N* in (1) in the Internet where LSPs may involve many hops. Again, each attempt has the same probability of success, and restorability in terms of the probability of eventual restoration within *M* attempts is (2) where P_1 is now (13).

In a successful restoration attempt, a RR message will travel *H* hops with a total delay of Ht_p. On any unsuccessful attempt, the RR message will be rejected at the *i*th LSR along the backup route with probability $q(1-q)^{i-1}$. The total delay involved in that unsuccessful attempt will be $i(t_r + t_p)$. The mean total delay involved in each unsuccessful attempt is therefore

$$E(T_{rel}) = \sum_{i=1}^{H} i(t_r + t_p) q (1-q)^{i-1} = (t_r + t_p)\left[1 - (1-q)^H (1 + Hq)\right] / q \tag{14}$$

The probability of exactly *m* unsuccessful attempts before a successful restoration is $P_1(1-P_1)^m$, $0 \le m < M$, and the mean number of unsuccessful attempts E(m) is given by (4) where P_1 is now (13). Combining (4) and (14), the total mean restoration time including Fault Notification message and failed attempts is

$$E(T_r) = t_d \sum_{h=1}^{H} h / H + Ht_p + E(m)E(T_{rel}) \tag{15}$$

where $E(T_{rel})$ is given by (14).

3. ADAPTIVE SELECTION OF BACKUP ROUTES

In the restoration approaches described above, the backup route is pre-established at the same time that the active path is established, but bandwidth is not reserved on the backup route in order to avoid restricting the amount of admissible traffic unnecessarily. Consequently, the selected backup path may not be the best choice after traffic conditions have had time to change. We propose an adaptive approach to periodically update the backup path selections. This process is carried out periodically in two steps: (1) LSRs will share up-to-date status and QoS information through extended OSPF, and (2) best candidate backup paths are recalculated. By default, the recalculations are performed at regular intervals but routing information updates advertising significant traffic changes can trigger an earlier recalculation. If the recalculated backup paths are different from the ones already established, the LSR routing tables are updated with the new backup paths. The objective is to maximize the probability of successful fault restoration on the first attempt, and thereby maximize restorability and minimize the mean restoration time for a fixed amount of network resources.

3.1. BACKUP PATH SELECTION

Searching for an optimal backup LSP is not a simple task because of the time varying rates of packet flows. The bandwidth requirements of a variable bit-rate packet flow is not straightforward to calculate. More bandwidth will result in better QoS, while less bandwidth will allow more traffic to be restored using a given amount of backup resources. The QoS will also depend on the amount of buffer space at each LSR and the traffic characteristics of other packet flows. The problem is made more difficult by the requirement to reroute all failed FECs simultaneously. Backup routes must be selected optimally to make the most use of a fixed set of available resources.

To overcome the first difficulty, the equivalent bandwidth concept is used to reduce the requirements of each FEC solely into terms of bandwidth. Equivalent bandwidth is a reversal of queueing analysis: given traffic characteristics and QoS requirements, the minimum required resources are calculated. More specifically, we used the equivalent bandwidth formulation that finds the required bandwidth given the source traffic peak rate *P*, mean burst period *b*, utilization factor ρ, buffer size *B*, and target packet loss ratio ε [25]. The delay requirement is considered later as an additional constraint. For convenience, the equivalent bandwidth is rewritten here as

$$c = P\frac{a-1+\sqrt{(a-1)^2+4\rho a}}{2a} \tag{16}$$

where $a = -\frac{b}{B}(1-\rho)\ln\varepsilon$. We note that the equivalent bandwidth for an FEC will vary on different links because of the dependence on an LSR's buffer space. This fact complicates the usual capacity optimization problem encountered in connection-oriented networks, where a fixed amount of bandwidth is required end-to-end for each flow.

Using the notion of equivalent bandwidth to translate the overall requirements of each FEC into bandwidth terms, an arc-path model can be developed for optimization to simultaneously route all backup LSPs satisfying their QoS requirements while efficiently utilizing available spare capacity [26,27]. Let G=[N,E] denote a network with node set N={1,2,…,n} and link set $E=\{e_1,e_2,\ldots,e_m\}$ of ordered pairs of nodes. An ordered pair of nodes is $e_m=(i,j)$ with $i\neq j$ and $i,j\in N$. A path in [N,E] with origin i_1 and destination i_{p+1} is a sequence $\{i_1,e_{m1},i_2,e_{m2},\ldots,i_p\}$. Let o_r and d_r denote the origin and destination for a demand $r\in\{1,2,\ldots,\gamma\}$.

The delay of a path is calculated as the sum of maximum queueing delays, i.e., buffer size divided by link transmission rate. Let D denote the maximum delay permitted for any backup LSP, and let P_r denote the set of backup LSPs whose delay is less than or equal to D. Let $Q_{re}\subset P_r$ denote the subset of paths in P_r containing link e. Let y_e denote the actual capacity of link e and F_e denote the set of demands affected by the failure of link e. For each $p\in P=\bigcup_{r=1,2,\ldots,\gamma} P_r$, let the variable c_{ip} equal the equivalent bandwidth in (16) if $i\in p$, and $c_{ip}=0$ otherwise. Let the binary variable g_{ep}=1 if path $p\in P$ is a backup LSP when e fails and $g_{ep}=0$ otherwise. When link e fails, we wish to determine backup LSPs for each demand in F_e such that the total equivalent capacity for each link in these LSPs is less than or equal to the actual capacity. Mathematically, we seek a set of binary variables, g_{ep}, such that

$$\sum_{p\in P} c_{ip} g_{ep} \le y_i \quad \forall i\in E \tag{17}$$

and

$$\sum_{p\in P_r\setminus Q_{re}} g_{ep} = 1 \qquad \forall r\in F_e \tag{18}$$

Since our goal is to minimize the total equivalent capacity needed for restoration, the optimization objective can be written as

$$\text{minimize} \sum_{i \in E} \sum_{p \in P} c_{ip} g_{ep} \tag{19}$$

Because of the computational difficulty, we rely on the CPLEX optimization software for solution.

4. AN EXAMPLE CASE STUDY

To demonstrate the spare capacity allocation procedure using the arc-path based optimization model for guaranteed service, the small MPLS network example illustrated in Figure 4 was solved. Table A describes the working LSPs, which include routing information and traffic characteristics for each working LSP. The link capacity and delay for traveling from one LSR to the next are shown in Table B. The buffer capacity for every LSR in the network is given in Table C.

Using this known information, the arc-path based optimization model was derived. The arc path model includes all feasible backup LSPs for all working LSPs in the network using the equivalent bandwidtch concept. Choosing an optimal backup LSP for each working LSP was accomplished using the CPLEX optimization system. For each failure *e*, (17)-(19) was solved using CPLEX 6.0 to obtain the optimal backup LSPs shown in Table D. For the QoS restrictions given (packet loss ratio and total delay), the paths found have the absolute minimum total equivalent capacity.

Due to the structure of the mathematical model, each link failure requires solving a simple discrete optimization problem having $m + \gamma$ constraints and $|P|$ binary variables.

Realistic problems having 30 nodes and 45 edges are tractable using the latest version of CPLEX. In previous work involving QoS, searching for backup routes can be very complex, but our strategy exploits equivalent bandwidth and well-known optimization techniques to reduce the complexity of the problem.

5. CONCLUSIONS

The described techniques for link restoration, partial path restoration, and path restoration attempt to minimize restoration time by distributing protocol functions to LSRs and pre-establishing backup routes. The different techniques have trade-offs in flexibility, restorability, and restoration time. Each approach can be improved by a proposed adaptive method to periodically recalculate and update optimal backup routes.

Spare capacity allocation is usually a difficult problem, and even more difficult with additional QoS constraints, but we reduce the complexity using equivalent bandwidth and arc-path based optimization models.

REFERENCES

[1] E. Rosen, A. Viswanatha and R. Callon, “Multi-protocol label switching architecture,” Internet draft draft-ietf-mpls-arch-05, Apr. 1999.

[2] R. Callon, et al., “A framework for multi-protocol label switching,” Internet draft draft-ietf-mpls-framework-02, Nov. 1997.

[3] B. Davie, P. Doolan, and Y. Rekhter, *Switching in IP Networks*, Morgan Kaufmann, 1998.

[4] A. Viswanathan, N. Feldman, Z. Wang and R. Callon, “Evolution of multi-protocol label switching,” *IEEE Commun. Mag.*, vol. 36, pp. 165-173, May 1998.

[5] Y. Katsube, K. Nagami, and H. Esaki, “Toshiba’s router architecture extensions for ATM: overview,” Internet RFC 2098, Apr. 1997.

[6] P. Newman, et al., “Ipsilon flow management protocol specification for IPv4 version 1.0,” Internet RFC 1953, May 1996.

[7] Y. Rekhter, et al., “Tag switching architecture overview,” *Proc. IEEE*, vol. 82, pp. 1973-1983, Dec. 1997.

[8] T. Li, “MPLS and the evolving Internet architecture,” *IEEE Commun. Mag.*, vol. 37, pp. 38-41, Dec. 1999.

[9] D. Awduche, “MPLS and traffic engineering in IP networks,” *IEEE Commun. Mag.*, vol. 37, pp. 42-47, Dec. 1999.

[10] Ghanwani, et al., “Traffic engineering standards in IP networks using MPLS,” *IEEE Commun. Mag.,* vol. 37, pp. 49 -53, Dec. 1999.

[11] K. Van Der Wal, M. Mandjes, and H. Bastiaanse, “Delay performance analysis of the new Internet services with guaranteed QoS,” *Proc. IEEE*, vol. 85, pp. 1947-1957, Dec. 1997.

[12] T. Chen and T. Oh, “Reliable services in MPLS,” *IEEE Commun. Mag.*, vol. 37, pp. 58-62, Dec. 1999.

[13] E. Crawley, et al., “A framework for QoS-based routing in the Internet,” Internet RFC 2386, Aug. 1998.

[14] R. Guerin, A. Orda, and D. Williams, “QoS routing mechanisms and OSPF extensions,” GLOBECOM’97, pp. 1903-1908, Nov. 3-8, 1997.

[15] L. Andersson, et al., “LDP specification,” Internet draft draft-ietf-mpls-ldp-05, June 1999.

[16] B. Jamoussi, ed., “Constraint-based LSP setup using LDP,” Internet draft draft-ietf-mpls-cr-ldp-02, Aug. 1999.

[17] D. Awaduche, et al., “Extensions to RSVP for LSP tunnels,” Internet draft draft-ietf-mpls-rsvp-lsp-tunnel-00, Nov. 1998.

[18] B. Davie, et al., “Use of label switching with RSVP,” Internet draft draft-ietf-mpls-rsvp-00, Mar. 1998.

[19] P. Demeester, et al., “Resilience in multilayer networks,” *IEEE Commun. Mag.*, vol 37, pp. 70 –76, Aug. 1999.

[20] R. Kawamura, K-I. Sato, and K. Tokizawa, “Self-healing ATM networks based on virtual path concept,” *IEEE J. Sel. Areas in Commun*., vol. 12, pp. 120-127, Jan. 1994.

[21] R. Kawamura and I. Tokizawa, “Self-healing virtual path architecture in ATM networks,” *IEEE Commun. Mag.*, vol. 33, pp. 72-79, Sep. 1995.

[22] P. Veitch and D. Johnson, “ATM network resilience,” *IEEE Network*, vol. 11, pp.26-33, Sept.-Oct. 1997.

[23] H. Komine, et al., “A distributed restoration algorithm for multiple-link and node failures of transport networks,” *GLOBECOM’90*, pp. 459-463, 1990.

[24] C. Chow, et al., “A fast distributed network restoration algorithm,” *12th Int. Phoenix Conf. Comp. Commun.*, pp. 261-267, 1993.

[25] R. Guerin, H. Ahmadi, and M. Naghshineh, “Equivalent capacity and its application to bandwidth allocation in high-speed networks,” *IEEE J. Selected Areas in Commun.*, vol. 9, pp. 968 –981, Sep. 1991.

[26] M. MacGregor and W. Grover, “Optimized k-shortest paths algorithm of facility restoration,” *IEEE Trans. Commun*., Sept. 1992.

[27] J. Kennington and M. Lewis, "A comparison of arc-path and node-arc formulations for allocating spare capacity to achieve optimal quality of service," SMU Dept. of Comp. Science Engin., Technical Report 00-CSE-1, Jan. 2000.

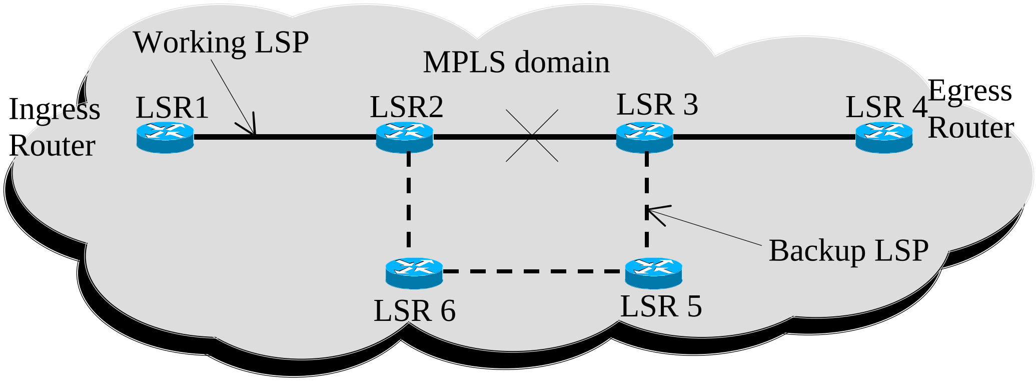

Figure 1. Link restoration.

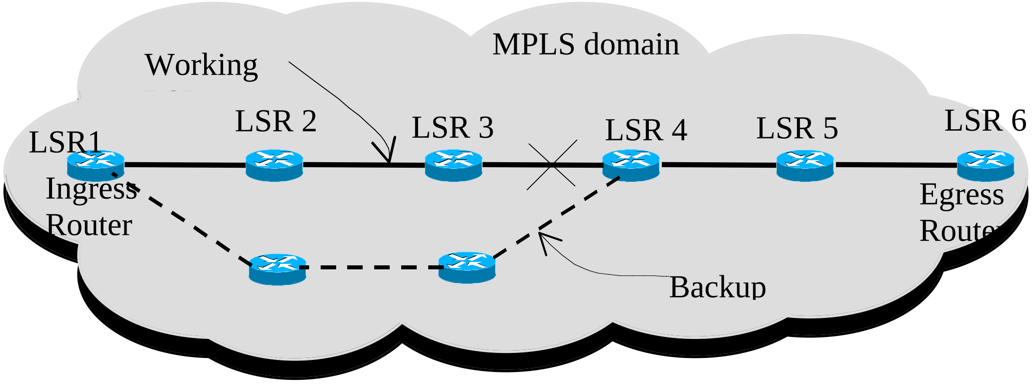

Figure 2. Partial path restoration.

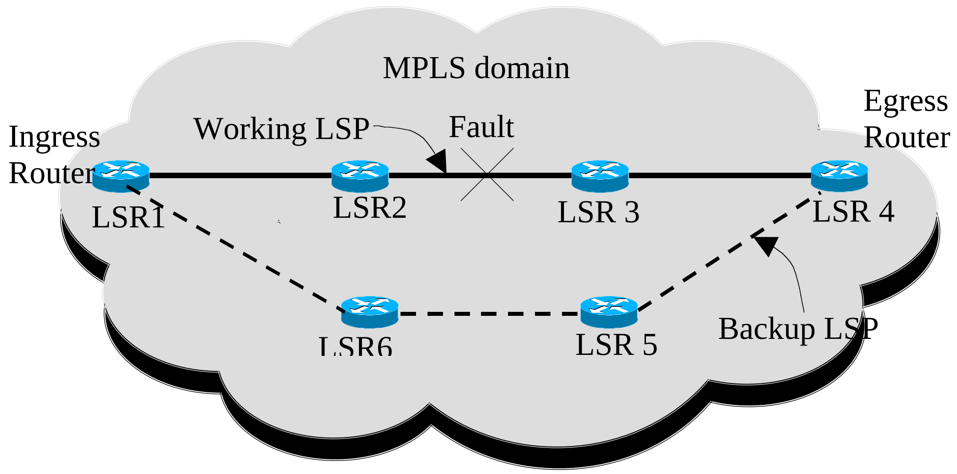

Figure 3. Path restoration.

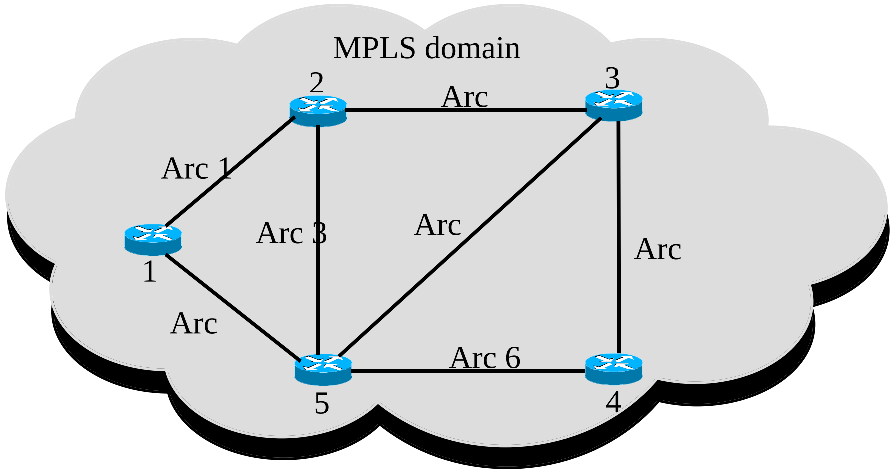

Figure 4. Example MPLS network for case study.

Table A. Example network: working LSPs and traffic characteristics

Working LSP #	Origin-destination	LSP	Peak rate	Average rate	Burst size	Loss prob.	Total delay
1	1-4	1-5-3-4	6 Mb/s	3 Mb/s	6 Mb	10^{-6}	5
2	2-5	2-3-4-5	7 Mb/s	5 Mb/s	6 Mb	10^{-6}	5
3	1-3	1-5-4-3	9 Mb/s	2 Mb/s	7 Mb	10^{-6}	5

Table B. Link capacities and delays

Link	Capacity	Delay
Arc 1 (1-2)	15 Mb/s	1
Arc 2 (1-5)	17 Mb/s	2
Arc 3 (2-5)	20 Mb/s	1
Arc 4 (2-3)	22 Mb/s	2
Arc 5 (3-5)	19 Mb/s	1
Arc 6 (4-5)	18 Mb/s	1
Arc 7 (3-4)	18 Mb/s	2

Table C. Buffer capacity at each LSR

LSR	Buffer size
1	6 Mb
2	10 Mb
3	5 Mb
4	4 Mb
5	7 Mb

Table D. Backup LSPs for any link failure

Failed link	Affected working LSP #	Optimal backup LSP	Total capacity	End-to-end delay
Arc 2	1 (1-5-3-4)	1-2-3-4	16.6 Mb/s	5
	3 (1-5-4-3)	1-2-3	16.4 Mb/s	3
Arc 4	2 (2-3-4-5)	2-5	6.3 Mb/s	1
Arc 5	1 (1-5-3-4)	1-2-3-4	16.6 Mb/s	5
Arc 6	2 (2-3-4-5)	2-5	6.3 Mb/s	1
	3 (1-5-4-3)	1-2-3	16.4 Mb/s	3
Arc 7	1 (1-5-3-4)	1-2-5-4	16.4 Mb/s	3
	2 (2-3-4-5)	2-5	6.3 Mb/s	1
	3 (1-5-4-3)	1-2-3	16.4 Mb/s	3