Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks

Yijun Xiong, Member, IEEE, and Lorne G. Mason

Abstract—This paper studies the capacity and flow assignment problem arising in the design of self-healing asynchronous transfer mode (ATM) networks using the virtual path concept. The problem is formulated here as a linear programming problem which is solved using standard methods. The objective is to minimize the spare capacity cost for the given restoration requirement. The spare cost depends on the restoration strategies used in the network. In this paper, we compare several restoration strategies quantitatively in terms of spare cost, notably: global versus state-dependent versus state-independent restoration. The advantages and disadvantages of various restoration strategies are also highlighted. Such comparisons provide useful guidance for real network design. Further, a new heuristic algorithm based on the minimum cost route concept is developed for the design of large self-healing ATM networks using path restoration. Numerical results illustrate that the heuristic algorithm is efficient and gives near-optimal solutions for the spare capacity allocation and flow assignment for tested examples.

Index Terms—ATM, heuristics, linear programming, network design, network reliability/survivability, self-healing.

NOMENCLATURE

The following notations are used in our work.

- $\mathcal{N}$: Set of nodes of the network.
- $\mathcal{A}$: Set of directed arcs of the network.
- $\mathcal{L}$: Set of links of the network, each link $l \in \mathcal{L}$ is composed of two arcs $(a, d \in \mathcal{A})$ which have the same end nodes as $l$ but with opposite directions.
- $\Pi$: Set of origin-destination node pairs (commodities).
- $O(\pi)$: Origin node of commodity $\pi$. $\pi \in \Pi$.
- $D(\pi)$: Destination node of commodity $\pi$. $\pi \in \Pi$.
- $\mathcal{S} \sim s_0$: Set of failure states of the network ($s_0$ is the nonfailure state).
- $R^c_\pi$: Set of candidate routes for commodity $\pi \in \Pi$ when the network is in state $s \in \mathcal{S}$.
- $\gamma^c_\pi$: Traffic demand expressing minimal bandwidth requirement for commodity $\pi \in \Pi$ when the network is in state $s \in \mathcal{S}$.
- $x^c_{\pi r}$: (Normalized) bandwidth used by commodity $\pi$ on route $r$ when the network state is in $s$, $\pi \in \Pi$, $r \in \mathcal{R}^c_{\pi}$, and $s \in \mathcal{S}$.
- $\delta_{N\omega}$: Delta function which equals 1 when network component $\omega$ is on route $r$ and 0 otherwise, $\omega \in \{\mathcal{N}, \mathcal{A}\}$.
- $\mathcal{F}(s)$: Set of failed components when the network is in state $s \in \mathcal{S}$.
- $c_\alpha$: Capacity on arc $\alpha \in \mathcal{A}$.
- $d_\alpha$: Length of arc $\alpha \in \mathcal{A}$.

I. INTRODUCTION

To quickly restore affected traffic upon network failure, typically within two seconds [1], sufficient redundant (spare) capacity must be preallocated in the network. The allocation of spare capacity will largely depend on the restoration scheme used, which is responsible for spare capacity searching and traffic rerouting after failure occurs. In general, less sophisticated restoration schemes (and shorter restoration time) will lead to allocating more spare capacity. To reach higher network utilization, more sophisticated restoration schemes are required. The spare capacity can be dedicated or shared among affected traffic of nonsimultaneous failures. Dedicated 1:1 (or 1 + 1) automatic protection switching (APS) is probably the simplest restoration scheme, but requires 100% redundancy (for 1 + 1, information is sent on two link/node disjoint paths and the APS is only done at the destination). This scheme has been successfully used in ring-type networks [2]. However, to use it in the mesh-type long-haul networks would be very costly, unless extremely fast restoration is required. Here we consider self-healing mesh-type networks with shared spare capacity.

Restoration schemes used in self-healing networks can in general be classified into two categories: reactive and preplanned. The flooding algorithm [3] and its modified versions [4], [5], designed for self-healing synchronous transfer mode (STM) networks are typical examples of the reactive restoration scheme, which start to search for alternate routes with sufficient spare capacity after failure occurs by broadcasting restoration messages. In contrast, for the preplanned restoration scheme, all restoration routes with sufficient bandwidth are precomputed by, for instance, the network management center for given failure scenarios and are downloaded to corresponding network nodes. In case of failure, a node responsible for restoring affected traffic simply activates the restoration routes and reroutes the affected traffic. The backup virtual path (VP) approach proposed in [6] for self-healing
asynchronous transfer mode (ATM) networks is an example of this scheme. Preplanned restoration can be used in the network environment where traffic patterns do not change frequently, e.g., in transport networks. However, for networks with rapid change of traffic patterns, information updating and route precomputing would become a burden. In such situations, reactive restoration may be more suitable as it requires little knowledge about current status of the network.

In this paper, we consider preplanned restoration schemes in the design of self-healing ATM networks under a single link or node failure scenario, with restoration performed on the VP level. The design objective is to provide the network survivability cost-effectively. Assuming that spare capacity is the main cost, the problem of designing restorable ATM networks thus becomes the optimization of capacity allocation and flow assignment to minimize the spare capacity cost while meeting the network survivability requirements. One purpose of this paper is to compare various restoration strategies (to be described below) for self-healing networks quantitatively in terms of spare cost. Another purpose is to present an efficient heuristic algorithm for the near-optimal design of large self-healing ATM networks.

The rest of the paper is organized as follows. Section II describes different restoration strategies for self-healing ATM networks and summarizes the previous work. The network design problem is then formulated and solved in Section III. Various restoration strategies are compared in Section IV in terms of spare capacity requirement (SCR). The efficient heuristic design algorithm is presented in Section V. The survivability in existing networks are examined in Section VI. Some concluding remarks are given in Section VII.

II. RESTORATION STRATEGY CONSIDERATION

In preplanned restoration schemes, restoration can be performed on link, fragment, and path basis (see Fig. 1). For path restoration, it can be classified into failure-oriented reconfiguration and global reconfiguration. In failure-oriented reconfiguration, only the affected working VP’s are rerouted, while in global reconfiguration, the whole layout of working VP’s (affected and unaffected) may be rearranged to overcome a failed link or node. It is obvious that global reconfiguration will require less spare capacity than failure-oriented reconfiguration, but it is probably not very practical if a restoration time limit is imposed. Nevertheless, it is worthwhile to consider global reconfiguration as it gives us the minimum spare capacity cost for the given restoration requirements.

Failure-oriented reconfiguration can be further divided into state-independent (SI) and state-dependent (SD) restoration. Here, the “state” means network failure state, indicating the failed network component(s). The backup VP approach proposed in [6] belongs to SI restoration, where for each working (primary) VP carrying real traffic there is a corresponding backup (secondary) VP which takes a link or node disjoint path. In case the working VP fails, the backup VP is activated by seizing the necessary spare bandwidth, and the traffic of the failed working VP is switched to the backup VP, regardless of the network failure state. It is like 1:1 APS, except that the spare capacity on a network link is shared among all backup VP’s passing through that link. Note that for the working VP’s associated with important services which require prompt restoration (much less than 2 s), dedicated 1 + 1 APS is likely to be used [8].

In SD restoration, the selection of restoration routes for an affected working VP will depend on network failure states. In other words, each working VP may have more than one “backup VP,” and the choice of a particular one in case the working VP fails will depend on the failure case. As the SD restoration is tailored to specified failures, normally it requires less spare capacity than the SI restoration. Note that link restoration could be regarded as a special case of both SD and SI restorations.

The appropriate restoration strategy ultimately applies in the design of self-healing networks will depend on many factors besides the spare capacity cost. Among them, restoration speed, VPI redundancy, and nodal processing/memory requirement are important criteria [7]. The tradeoff between these factors was discussed in [9]. Here we compare various restoration strategies in terms of spare capacity cost. The comparison of SI and SD restorations in terms of the above three criteria is given in [10].

Compared to reactive restoration schemes, preplanned restoration schemes are less flexible in handling unpredicted failure scenarios. To cope with unexpected failures, there are two possible approaches: 1) the remaining affected traffic, which cannot be restored by a preplanned scheme, is restored by the network management center (centralized control) and 2) using flooding algorithms to restore the remaining affected traffic.

Relation to Previous Work: The design of self-healing mesh-type networks is an optimization problem, i.e., optimizing capacity allocation and flow assignment to minimize the cost, which can be formulated as a (pure/mixed) integer programming (IP) and approximated by linear programming (LP). This problem was studied before, e.g., in [11]–[15] for link restoration and in [16] and [17] for global reconfiguration with path restoration. Using the multicommodity flow model, papers [11], [13], and [14] formulate the problem as an LP. The algorithms described in [11], [13], and [14] give optimal solutions for spare capacity allocation. However, the corresponding optimal flow assignments are only available
in [13] and [14], not in [11]. Node failure scenarios and hop-limit constraints were considered in [14]. A practical and near-optimal algorithm was developed in [12] for spare capacity allocation, which uses k-shortest link disjoint paths for traffic rerouting [15]. The above problem was analyzed in [16] as a nonsimultaneous multicommodity flow model. An efficient suboptimal solution procedure was presented which is suitable for large-scale networks; but, as in [11], no flow assignment is provided. [17] formulated the same problem as an IP. The upper and lower bounding technique used in [17] can be applied to small- and moderate-scale networks. Here we mainly consider failure-oriented path restoration. Related work can be found in [18]–[20]. The SD path restoration was studied as an IP in the context of STM in [18]. The SI path restoration with spare optimization was studied as a mixed IP (integer valued link capacity and bifurcated restoration flow) in [19]. A comparison of link and path restorations in terms of restoration ratio was given in [20]. Unlike [18]–[20], we provide a unified modeling approach and compare global versus failure-oriented reconfiguration and SI versus SD restoration in regard to spare cost under link and node failure scenarios. Further, a heuristic is developed for the SD and SI restorations in large-scale self-healing networks. Other related work can be found in [21]–[23], where [21] studied the circuit-switched network survivability by considering the teletraffic network and the underlying facility network together, [22] studied various ATM self-healing design options and route planning, and [23] considered dynamic VP bandwidth allocation.

III. NETWORK DESIGN: A THEORETICAL APPROACH

Consider a network \( G(\mathcal{N}, \mathcal{L}) \) which has \(|\mathcal{N}|\) nodes and \(|\mathcal{L}|\) links, where \(|\mathcal{N}|\) is the cardinality of set \( \mathcal{N} \). In this network, each of its components (links and nodes) has two states: normal state 0, and failure state 1. The network state is completely described by a vector of these component states. We use \( \mathcal{S} \) to denote the set of network states. Denoting by \( s_0 \) the network normal operating state (no failure), \( \mathcal{S} - s_0 \) is the set of network failure states. For a single failure scenario, \(|\mathcal{S}| = |\mathcal{L}| + 1\). For a single or node failure scenario, \(|\mathcal{S}| = |\mathcal{L}| + |\mathcal{N}| + 1\). To be general, the bidirectional link case is considered.

We use static bandwidth allocation to VP’s. The bandwidth of each VP is determined by its traffic type and the quality of service requirement [24]. The total bandwidth between a node pair is equal to the sum of individual VP bandwidth between that node pair. To simplify the survivable network design problem, the traffic flow in the network is modeled as a multicommodity flow. Complete (100%) restoration at failure events is considered.

**Assumptions:** The network design problem in general can be formulated as an IP which is NP-hard. To ease the computational problem and allow us to investigate relatively large networks, it is assumed that no capacity modularization will be made to conform to physical transmission systems, i.e., the link cost is proportional to the link length and the capacity of the link.

For VP restoration, one may consider 1) bifurcated flow restoration, where the traffic flow of a failed working VP may be restored by one or more than one VP and 2) nonbifurcated flow restoration, where the traffic flow of a failed working VP can only be restored by one (backup) VP. The design problem is a mixed IP when using nonbifurcated flow restoration but reduces to LP if using bifurcated flow restoration. In this and the next sections, we mainly assume bifurcated flow restoration, although the formulations can be easily extended to the case of nonbifurcated flow restoration. The nonbifurcated flow restoration will be considered in Section V.

The above two assumptions isolate the influence of restoration strategies from various effects due to modularity and detailed VP bandwidth allocation, etc., which allow us to compare different restoration strategies in a wide range of design and failure scenarios. The corresponding LP solution will give a lower bound on network cost for the nonbifurcated flow case which can also serve as a reference in comparing the efficiency of heuristic design algorithms. Further, we have observed from LP solutions that in most cases there exist only a few optimal restoration routes for traffic demand between each node pair. As there could be many VP’s of various bandwidth between a node pair, the LP solution might be a good approximation for the nonbifurcated VP flow restoration. Even in cases where the traffic flow of a failed VP has to be bifurcated for restoration, the integrality of VC’s (virtual channels) may not be affected as the bandwidth of a VC is usually much smaller than the bandwidth of a VP.

While the LP is one extreme, another extreme is to consider the traffic flows of VP’s between a node pair as the traffic flow of one (super) VP. Obviously, the corresponding IP solution will give an upper bound on network cost. As one will see in Section IV-D, the upper and lower bounds are actually very close. So the comparative results based on LP are expected to be valid even if the bifurcated flow assumption is removed.

A. Path Restoration

1) Global Reconfiguration: For global reconfiguration, in case of failure, all traffic flows, affected and unaffected, are reconfigured so that the restoration ratio is guaranteed but the total network cost is minimized. The survivable network design is formulated as the following LP problem.

**F1:**

\[
\text{minimize} \quad \sum_{a \in \mathcal{A}} d_a c_a \\
\text{subject to} \quad \sum_{r \in R_a^s} x_{r,a}^s = 1, \quad \pi \in \Pi, \quad s \in \mathcal{S} \quad (1) \\
\quad c_a \geq \sum_{\pi \in \Pi} \sum_{r \in R_a^s} \delta_{r,a} \gamma_{r,a} x_{r,a}^s, \quad a \in \mathcal{A}, \quad s \in \mathcal{S} \quad (2) \\
\quad x_{r,a}^s \geq 0, \quad r \in R_a^s, \quad \pi \in \Pi, \quad s \in \mathcal{S} \quad (3)
\]

In the above formulation, the constraints in (1) and (3) guarantee that the traffic demand \( \gamma_{r,a} \) between each node pair is satisfied in every possible network state. Constraints (2) ensure that the capacity assigned to arc \( a \) is large enough to accommodate the traffic flows on arc \( a \) for all possible network states. The number of constraints in formulation F1
are roughly equal to
\[ N_s(F_1) \approx |\Pi||S| + |A||S| \]  
(4)

where \(|\mathcal{X}|\) is the cardinality of the set \(\mathcal{X}\).

Denoting by \(s_0\) the nonfailure state, we usually have \(\gamma^{-}_\pi \leq \gamma^{+}_\pi\), for \(\pi \in \Pi\) and \(s \in S - s_0\). Let \(\mathcal{O}(\pi)\) and \(\mathcal{D}(\pi)\) represent the origin and destination nodes of commodity \(\pi \in \Pi\) respectively. Obviously, \(\gamma^{\pm}_\pi = 0\) if \(\mathcal{O}(\pi)\) or \(\mathcal{D}(\pi)\) \(\notin \mathcal{F}(s)\) because a failed node cannot send or receive any information. Let \(\Pi^s\) be the subset of commodities, each of which cannot be restored due to the failure of origin or destination node of the commodity. For complete restoration \(\bigcup_{\pi \in \Pi} \gamma^s_\pi = \bigcup_{\pi \in \Pi^s} \gamma^s_\pi\).

Formulation \(F1\) (and \(F2\) below) can be easily modified to handle nonbifurcated flow case. Suppose each node pair has only one VP with bandwidth \(\gamma^{\pm}_\pi\). The IP formulation for nonbifurcated flow restoration is the same as \(F1\) except replacing (3) by the constraints
\[ x^s_{\pi r} \in \{0, 1\}, \quad r \in R^s_\pi, \quad \pi \in \Pi, \quad s \in S. \]

The extension to more than one VP per commodity is straightforward. Suppose a node pair \(\pi\) has \(V^\pi\) VPs, \(\pi \in \Pi\), the number of constraints in the IP formulation is roughly equal to \(|S|\sum_{\pi \in \Pi} V^\pi + |A||S|\). If the average number of VP’s per node pair is \(\bar{V}\), the number of constraints (as well as variables) is almost \(V\) times that in the LP formulation \(F1\).

2) Failure-Oriented Reconfiguration: In failure-oriented reconfiguration, only affected traffic flows are rerouted at failure events. Unaffected traffic flows will remain unchanged. To be clear, here we use \(y^s_\pi\) instead of \(x^s_\pi\) to denote a restoration flow on route \(r\) for the affected commodity \(\pi\) when the network is in failure state \(s\), \(r \in R^s_\pi, \pi \in \Pi, s \in S - s_0\). As above, \(x^s_{\pi r}\) is a (normalized) flow of commodity \(\pi\) using route \(r\) when the network is in nonfailure state \(s_0\), \(r \in R^0_\pi\).

The network design based on failure-oriented reconfiguration can be formulated as the following LP problem.

\[ F2: \]

\[ \text{minimize} \quad \sum_{a \in A} \left( \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} \gamma^{\pm}_\pi x^0_{\pi r} + c^\text{spare}_a \right) \]

subject to
\[ \sum_{r \in R^0_\pi} x^0_{\pi r} = 1, \quad \pi \in \Pi \]
\[ c^\text{spare}_a \geq \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} f^s_{\pi r} y^s_\pi, \quad a \in A, s \in S - s_0 \]
\[ \sum_{r \in R^0_\pi} y^s_\pi = \sum_{r \in R^0_\pi} \Omega_{\pi r s} x^0_{\pi r}, \quad \pi \in \Pi, s \in S - s_0 \]

where \(f^s_{\pi r}\) is the restoration level \((0 \leq f^s_{\pi r} \leq 1\), \(x^0_{\pi r} \geq 0, y^s_\pi \geq 0, \Omega_{\pi r s} = 1\) if \(F(s) \cap \{r\} \neq \emptyset\) and otherwise 0. For complete restoration \(f^s_{\pi r} = 1\). In the above formulation, the first term in the objective function is the capacity required on arc \(\alpha\) to carry traffic in nonfailure state \(s_0\). The second term \(c^\text{spare}_a\) represents the additional (spare) capacity needed to restore affected traffic in case of failure. The sum of these terms gives the total capacity requirement on arc \(\alpha, \alpha \in A\). Constraints (7) ensure that for each commodity \(\pi\), its (normalized) affected traffic flows are restored in every possible failure case. The number of constraints in the above formulation is almost the same as that in formulation \(F1\) given in (4). However, the constraint matrix in \(F2\) is less sparse than that in \(F1\) which means more computation time is required to solve \(F2\).

In formulation \(F2\), each affected traffic flow will take alternative routes in case of failure [constraints (7)]. But the capacity of the route used by the affected traffic flow before failure occurs is not released for restoration [constraints (6)], even though some network components along the route do not fail. It is clear that the SCR can be reduced by making full use of that capacity. Suppose when failure occurs the capacities used by affected traffic flows are released before restoration, we get another formulation \(F2 - m\), which is the same as \(F2\) except that constraints (6) are now replaced by
\[ c^\text{spare}_a \geq \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} f^s_{\pi r} y^s_\pi, \quad a \in A, s \in S - s_0. \]

Releasing capacities occupied by the affected traffic was also considered in [18] and is called \(\text{stub release}\) there. Note that formulations \(F1, F2\), and \(F2 - m\) for path restoration are general and can cope with various failure scenarios such as any single failure or multiple failures case. For the nonbifurcated flow restoration, we simply impose constraints \(\sum_{r \in R^0_\pi} x^0_{\pi r} \leq 1\) if each commodity \(\pi\) contains only one VP.

B. Link Restoration

By link failure we mean all communication capacities between the two nodes directly connected by the failed link are completely lost (i.e., span failure). As depicted in Fig. 1(a), the two nodes directly connected by the failure link will be responsible for the restoration of traffic flows on that link. Similarly, we get the following formulation for the link restoration.

\[ F3: \]

\[ \text{minimize} \quad \sum_{a \in A} \left( \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} \gamma^{\pm}_\pi x^0_{\pi r} + c^\text{spare}_a \right) \]

subject to
\[ \sum_{r \in R^0_\pi} x^0_{\pi r} = 1, \quad \pi \in \Pi \]
\[ c^\text{spare}_a \geq \sum_{b \in F(s)} \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} y^s_{\pi b}, \quad a \in A, s \in S - s_0 \]
\[ \sum_{r \in R^0_\pi} y^s_{\pi b} = \sum_{\pi \in \Pi} \sum_{r \in R^0_\pi} \delta_{\pi a} f^s_{\pi r}, \quad b \in F(s), s \in S - s_0 \]
where \(x_{rb}^m \geq 0, \ y_{rb}^m \geq 0, \ f_{rb}^m \) is the restoration level for affected traffic of failed arc \(b\), \(0 \leq f_{rb}^m \leq 1\). Here, traffic flows on each arc \(b \in A\) are regarded as one commodity. \(R_b^s\) is now the set of candidate routes for restoring traffic flows on arc \(b\) when it fails in network state \(s \in S\). \(F(s)\) is the set of failed arcs in network state \(s\). As before, constraints (10) ensure that sufficient capacity is assigned to each arc to restore affected traffic at failure events. The right-hand side of (11) is a portion \((f_{rb}^m)\) of the aggregate traffic on arc \(b\). Constraints (11) assure that a part \((f_{rb}^m)\) of traffic flows on each arc will be restored in case the arc fails. The number of constraints in the above formulation is approximately equal to

\[
N_c(F3) \approx |\Pi| + |A||S| + \sum_{s \in S} |F(s)|. \tag{12}
\]

IV. NUMERICAL EXAMPLES

A. Network Description

The four network examples shown in Fig. 2 are used in the comparison of restoration strategies. Here, the notation \(\text{Net}_s(|N|, |L|)\) is used to represent \(s\)th network which has \(|N|\) nodes and \(|L|\) links. For example, \(\text{Net}_1(11, 23)\) stands for network 1, which has 11 nodes and 23 links. Networks 1 and 3 were used in \cite{13} and network 4 was given in \cite{3}. The link distances in networks 2 and 4 are just some arbitrary numbers drawn there. Note that networks 1 and 4 are dense networks which usually appear in Metropolitan Area Networks, while networks 2 and 3 are sparse networks which are often found in long-haul networks.

Both uniform and nonuniform traffic demands are considered in the numerical examples. In uniform traffic demands, the bandwidth requirement between each node pair equals \(B\) (bits per second) in the nonfailure state, i.e., \(\gamma_{\pi}^0 = B, \pi \in \Pi\). Two types of nonuniform traffic demands are selected.

Type I: Traffic demand between a node pair \(\pi\) decreases proportionally to the minimum hops \(H_{\pi}\) between the node pair, i.e., \(\gamma_{\pi}^0 = B/H_{\pi}\).

Type II: Traffic demand between a node pair is uniformly distributed between 0.1 and 1.9\(B\) with mean value of \(B\).
For nonuniform traffic demands, only a symmetric traffic case is considered where the two traffic demands between a node pair are equal to each other, i.e., $\gamma_\pi = \gamma_{\pi'}$, if $O(\pi) = D(\pi')$ and $D(\pi) = O(\pi')$. $\pi, \pi' \in \Pi$.

A single link or node failure scenario is considered here. A node failure is equivalent to simultaneous failures of all links connected to the node. However, it should be noticed that link failure scenarios cannot always be regarded as special cases of node failure scenarios, because some traffic demands will be lost in node failure scenarios which is not necessarily the case in link failure scenarios. We found that networks dimensioned with 100% restoration (of restorable affected traffic) for a single node failure scenario cannot always guarantee 100% restoration in a single link failure scenario.

When all commodities (working VP’s) in a network take the shortest paths to their destinations in the nonfailure state $\emptyset$, we denote by $C_\omega$, the resulting network cost. It is clear that $C_\omega$ is constant for the given network topology and traffic demand matrix. Based on working capacity cost $C_\omega$, the SCR of a self-healing network is defined as the ratio of spare capacity cost $C_s$ to the working capacity cost, i.e.,

$$\text{SCR} = \frac{C_s}{C_\omega} = \frac{C_t - C_\omega}{C_\omega} \quad (13)$$

where $C_t$ is the total network cost.

For the given network topology, traffic demands, and restoration scheme, the value of SCR also depends on how the network is optimized and which set of candidate routes between each node pair is used. Two kinds of network optimization are considered: 1) joint optimization of working and spare capacities and 2) optimization of spare capacity for the given working VP’s which use the shortest paths. Further, two sets of candidate routes are considered for each node pair. The first set contains all possible paths (APP) between a node pair, while the second set consists of only (mutually) link disjoint paths (LDP) between the node pair. So there are four combinations in network optimization: (j, APP), (s, APP), (j, LDP), and (s, LDP), where notation (j, APP) means joint optimization using APP, and (s, APP) stands for spare capacity optimization using APP, etc.

Let $R_\pi$ be the set of candidate routes for commodity $\pi \in \Pi$. To reduce the huge number of variables in formulations $F_1$, $F_2$, and $F_3$ when using APP, the maximum number of candidate routes between each node pair $\pi$ is restricted to a certain value, say $M$, i.e., $|R_\pi| \leq M, \pi \in \Pi$. It is clear that $R_\omega = R_\pi \subseteq R_\sigma$ for $\sigma \in \mathcal{S} - \emptyset_0$, due to network component failures. Further, hop limit $H$ is imposed in selecting candidate routes. So for APP, set $R_\pi$ actually contains $M$ shortest paths between node pair $\pi$ which satisfy the hop-limit requirement. In the following numerical results we set $M = 40$. For networks 1 and 2 in Fig. 2, we take $H = 6$ and for networks 3 and 4, $H = 10$. The candidate routes set $R_\pi$, $s \in \mathcal{S}$, is determined once APP or LDP is chosen. For spare optimization, as working VP traffic flows are known, constraints (5) in $F_2$, and (9) in $F_3$ can be removed.

B. Global Versus Failure-Oriented Reconfiguration

Let us first compare different restoration strategies based on networks 1 and 2, under two failure scenarios: any single link failure and any single link or node failure. The restoration ratio is 100%. Figs. 3 and 4 depict the SCR’s in networks 1 and 2, respectively, with uniform traffic demands and path restoration. It is observed that:

1) as expected, global reconfiguration ($F1$) requires less spare capacity than failure-oriented reconfiguration ($F2$);
2) the difference between $F1$ and $F2$ in SCR becomes larger as the optimization option changes from (j, APP) to (s, APP), i.e., failure-oriented reconfiguration is more sensitive to optimization options;
3) the SCR can be reduced if reusing the capacity occupied by the affected traffic ($F2 - m$);
4) joint optimization is better than spare optimization;
5) optimization using APP is much better than using only LDP, especially for $F2$, and (j, APP) yields the minimal SCR.
The above observations are understandable, as the more flexible the optimization procedure, the less the spare capacity is likely required. The penalty, of course, is the more sophisticated restoration algorithm.

An interesting phenomenon revealed in Figs. 3 and 4 is that the difference in SCR between global and failure-oriented reconfiguration is quite small when using (j, APP), although the difference is more pronounced in a sparse network and in case of node failure. This suggests that using failure-oriented reconfiguration $F_2$, which is more implementable than global reconfiguration $F_1$ when considering restoration time, the additional SCR is marginal (only a few percent more). Moreover, this additional cost could be reduced by reusing the capacity occupied by the affected traffic ($F_2-m$).

Similar results are also observed for nonuniform traffic demands. As an example, the average SCR’s in network 1 are depicted in Fig. 5 for nonuniform type I traffic demands. The SCR’s for nonuniform type II traffic demands are given in Table I. From now on, we will concentrate on uniform traffic demands only.

C. Path versus Link Restoration

Figs. 6 and 7 compare the link and path restorations on a dense and a sparse network, respectively. They illustrate that the path restoration ($F_2$) requires less spare capacity than the link restoration ($F_3$), which is understandable because link restoration is less flexible in selecting restoration routes and thus cannot better share the spare capacity. The situation becomes worse when the network is sparse and when using LDP. It is also observed that the difference of SCR is relatively small in dense networks, especially when using (j, APP).

As formulations $F_1$ and $F_2$ of path restoration can only be solved for small to moderate networks and formulation $F_3$ is able to solve large networks, the latter observation suggests that one could probably use $F_3$ to obtain an optimal solution for the link restoration and then convert it to a suboptimal solution for the path restoration. Moreover, since origin and destination node information of affected traffic flows are not considered in the link restoration, the restoration route for an affected traffic flow may even pass through the origin and/or destination nodes of that flow more than once (backhaul phenomenon [7]). For instance, for a traffic flow between node pair (1, 4) in network 1, in the nonfailure state it will take route 1-0-4. When the link connecting node pair (1, 0) fails, the optimal restoration routes obtained from $F_3$ is 1-
2-4-0. Hence, the traffic flow between (1, 4) in the failure case will take the route 1-2-4-0-4. Clearly, the loop between nodes 4 and 0 is unnecessary. To omit the back-haul part when forming path restoration routes from the optimal solution of \( F^3 \), a heuristic algorithm called \( F^3 - c \) is developed which is described below.

**Heuristic Algorithm (\( F^3 - c \))**:

**Step 0**: To obtain an optimal solution for the link restoration using \( F^3 \).

**Step 1**: Given a failure case, for each optimal restoration route, to assign it to an affected traffic flow which will yield the maximum hop savings after deleting the back-haul part.

**Step 2**: To repeat step 1 until all the affected traffic flows are restored.

**Step 3**: To repeat steps 1 and 2 until all failure cases are considered.

For example, assigning the optimal restoration route 1-2-4-0 to the affected traffic flow 1-0-4-0 in network 1 will save 2 hops. An alternative to step 1 is for each affected traffic flow to find an optimal restoration route which will give the maximum hop savings after deleting the back-haul part. It is found that in both cases, the results are very close to each other. Using the above heuristic algorithm, the new restoration route is 1-2-4, which is shorter and, hence, uses less spare capacity. Therefore, it is expected that after converting the optimal solution of \( F^3 \), the SCR will be reduced. Surprisingly, no reduction in SCR is obtained in network 1. However some reductions are observed in network 2 for \((s, APP)\) and \((s, LDP)\) as depicted in Fig. 7. The possible reason is the following. The spare capacity on each arc is determined by the restoration flows passing through that arc under all failure cases, so removing the back-haul part of restoration flows on the arc under one failure case does not necessarily lead to reduction of the final SCR on the arc. In sparse networks, as there are fewer alternative routes for restorations, removing the back-haul part might be helpful in reducing the spare capacity. From the above example, we also see that link restoration cannot guarantee an end-to-end hop limit for a traffic flow, even though there is a hop limit in selecting restoration routes.

The numerical results depicted in Figs. 3–7 are based on networks of relatively small size. To study the impact of network size on the SCR, we consider using networks 3 and 4 shown in Fig. 2. Very often, the connectivity of a network \( G(N, L) \) is measured by the network average node degree \( \bar{D} \), which is equal to the average number of links at each node. It is easy to see that \( \bar{D} = 2L/N \), where \( 1 \leq \bar{D} \leq N-1 \). In dense network 1, \( \bar{D} = 4.18 \) and in sparse network 2, \( \bar{D} = 3.1 \). For a given type of network, the values of \( \bar{D} \) should be close to each other. As the difference in SCR between \( F^1 \) and \( F^2 \) (also between \( F^2 \) and \( F^3 \)) is more pronounced in sparse networks than in dense networks, our numerical study is focused on sparse networks. Specifically, a sparse network \( Net_{3}(20, 30) \) with \( \bar{D} = 3.0 \) is used as an example, which is a subnetwork of network 3 and consists of nodes 0–19.

Fig. 8 illustrates the impact of network size on the SCR for sparse network \( Net_{3}(20, 30) \) under a single link failure scenario. Let us look at the optimization option \((j, APP)\). Fig. 8 shows that the difference between \( F^1 \) and \( F^2 \) in SCR is not very sensitive to the network size. In other words, using failure-oriented reconfiguration instead of global reconfiguration, the sacrifice in SCR might not be very significant in large networks. As the SCR will also depend on the network topology, it is difficult to predict what exactly the additional cost would be for a specific large network. One can also see from Fig. 8 that the difference in SCR between link and path restorations is quite sensitive to the network size, and therefore the additional spare cost of using the link restoration could be significant in large sparse networks. Further, heuristic algorithm \( F^3 - c \) for the path restoration becomes less efficient as the network size grows. Consequently, other efficient heuristic algorithms need to be developed for large networks. From the previous numerical examples we know that the gaps between \( F^1 \) and \( F^2 \) as well as between \( F^2 \) and \( F^3 \) are smaller in dense networks than in sparse networks, we found that this situation still holds in large dense networks.

**D. LP Versus IP**

As discussed in Section III, if we assume that the traffic demand \( \gamma_x \) of a node pair belongs only to one (super) VP and will be restored in an integral fashion when the VP fails, then the corresponding IP solution will give an upper bound on the SCR. We found that the IP and LP (lower bound) solutions are very close, as shown in Table II, for the example of failure-oriented restoration and \((s, APP)\). The explanation is that the spare capacity is shared by backup VP’s of nonsimultaneously failed working VP’s and a network failure will only affect a small portion of the working VP’s. This finding could be exploited to increase the utilization of self-healing ATM networks and to speed up the restoration process by restoring affected working VP’s between a node pair together [10].

**V. NETWORK DESIGN: A HEURISTIC ALGORITHM**

Here a new heuristic algorithm is proposed for designing large self-healing ATM VP networks which use failure-
TABLE II
SCR USING FAILURE-ORIENTED RESTORATION AND (s, A)(P) UNDER LINK FAILURE SCENARIO

<table>
<thead>
<tr>
<th>network</th>
<th>Net. 1(11,23)</th>
<th>Net. 3(20,30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization</td>
<td>LP</td>
<td>IP-LP</td>
</tr>
<tr>
<td>Uniform demand</td>
<td>56.46%</td>
<td>0</td>
</tr>
<tr>
<td>Nonuniform</td>
<td>51.30%</td>
<td>0.42%</td>
</tr>
</tbody>
</table>

oriented reconfiguration and path restoration. This algorithm is suitable for both SD and SI restorations with nonbifurcated flows. To facilitate the description, we still use the terminology backup VP to denote a restoration flow for a failed working VP. In the SI restoration, each working VP has only one backup VP, which takes a link/node disjoint route. In case the working VP fails, the affected traffic of this working VP will be carried by its backup VP. In the SD restoration, apparently, each working VP may have more than one backup VP and to use a particular backup VP in case the working VP fails will depend on the network failure state. It is assumed that working VP’s are given, which take the shortest routes between origin and destination node pairs. The heuristic algorithm uses the following minimum cost route (MCR) concept to minimize the spare capacity cost. To be clear, let us first consider the SD restoration. Suppose each commodity π is composed of \( V_{s} \) VP’s, and the bandwidth of \( i \)th VP is \( B_{\pi, i} \). \( 1 \leq i \leq V_{s}, \pi \in \Pi \). Let \( Q_{\pi, a}^{\text{spare}} \) be the spare capacity available on arc \( a \in A \). If \( i \)th working VP of commodity \( \pi \) fails when the network is in failure state \( s \), \( s \in S - s_{0} \), a route with minimum cost, denoted by \( \text{BVP}_{\pi, s}^{s} \), will be chosen as its backup VP, i.e.,

\[
\text{cost}(\text{BVP}_{\pi, s}^{s}) = \min_{v \in R_{\pi}^{s}} \left( \sum_{a \in A} \delta_{va} d_{a}(B_{\pi, i} - Q_{\pi, a}^{\text{spare}})^+ \right), \forall i, \pi \in \Pi, s \in S - s_{0} \tag{14}
\]

where \((\cdot)^+ = \max(\cdot, 0)\) and \( R_{\pi}^{s} \) is the set of candidate paths for commodity \( \pi \) in failure state \( s \). After choosing the MCR \( \text{BVP}_{\pi, s}^{s} \), the available spare capacity on each arc of \( \text{BVP}_{\pi, s}^{s} \), will be updated.

When the network is in each failure state \( s, s \in S - s_{0} \), only a portion of working VP’s are affected. The spare capacity \( c_{a}^{\text{spare}} \) on each arc \( a \in A \) is equal to the maximum of spare capacities required in all network failure states, i.e.,

\[
c_{a}^{\text{spare}} = \max_{s \in S - s_{0}} \{ c_{a}^{\text{spare}}(s) \}.
\]

In other words, the SCR on arc \( a \) is only determined by the worst failure state \( s^* \) which requires the maximum spare capacity on arc \( a \), i.e., \( c_{a}^{\text{spare}}(s^*) = c_{a}^{\text{spare}} \).

The main idea of the proposed heuristic algorithm is to find the worst failure state for a given arc and to minimize the spare capacity on that arc. A similar principle was used in [25] for link (span) restoration in STM networks. The above observation was made independent of [25]. Keeping this idea in mind, the new heuristic algorithm called MCR is described as follows.

**MCR Heuristic Algorithm:**

**Step 0:** Allocate each working VP along the shortest route and set arc spare capacity to zero \( c_{a}^{\text{spare}} = 0 \). Note that for SI restoration, the selected shortest route should have at least a single link/node disjoint route to be used by the backup VP.

**Step 1:** For every failure case, find the affected working VP’s. For each affected working VP (see option 1 below), find the MCR using formula (14), and update the arc spare capacity.

**Step 2:** For a given (tagged) arc (option 2): a) find the worst failure state \( s^* \) and the respective affected working VP’s; b) restore failed working VP’s if their backup VP’s do not pass the tagged arc; c) for each failed working VP with its backup VP passing the tagged arc, try to find a new backup VP not passing the arc but with zero additional spare cost; and d) if such a new backup VP is not available, then

\[
\text{if (option}_3 = 0) \quad /\ast \text{conditional MCR} /\ast
\]

using the original backup VP;

\[
\text{else}
\]

find a new backup VP using (14).

**Step 3:** Repeat step 2 until all arcs are considered.

**Step 4:** Repeat steps 2 and 3 until the total spare cost cannot be further reduced.

**Options:**

option 1: VP sequence (0) or commodity sequence (1);
option 2: arc sequence (0) or arc spare capacity decrease order (1);
option 3: conditional MCR (0) or unconditional MCR (1);
option 4: spare cost increase not allowed (0) or allowed (1).

In the above algorithm, steps 0 and 1 are initial assignment phase and steps 2–4 are modification phase. There are four options and in total 16 combinations. The commodity sequence in option 1 means working VP’s are handled in commodity order. The same algorithm is also applicable for the SI restoration. In this case, set \( R_{\pi}^{s} \) contains only those routes which are link/node disjoint to the shortest route used by the working VP’s between node pair \( \pi, \pi \in \Pi \). Further, since there is only one backup VP for each working VP.
TABLE III
A COMPARISON OF ALGORITHMS IN TERMS OF SCR’S IN NETWORKS WITH UNIFORM TRAFFIC DEMANDS, FIVE VP’S PER COMMODITY AND USING APP SETS (M = 40)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>SSCA</th>
<th>F3-c(s)</th>
<th>F3-c(j)</th>
<th>MCR-SI</th>
<th>MCR-SI</th>
<th>F2(s)</th>
<th>F2(j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net.(11,23)</td>
<td>80.36%</td>
<td>59.23%</td>
<td>55.41%</td>
<td>61.47%</td>
<td>57.50%</td>
<td>56.46%</td>
<td>54.44%</td>
</tr>
<tr>
<td>Net.(15,30)</td>
<td>78.66%</td>
<td>63.97%</td>
<td>44.17%</td>
<td>54.56%</td>
<td>50.94%</td>
<td>48.02%</td>
<td>36.41%</td>
</tr>
<tr>
<td>Net.(20,42)</td>
<td>77.50%</td>
<td>59.24%</td>
<td>43.56%</td>
<td>51.86%</td>
<td>48.52%</td>
<td>45.80%</td>
<td>34.56%</td>
</tr>
<tr>
<td>Net.(25,55)</td>
<td>71.22%</td>
<td>55.78%</td>
<td>39.70%</td>
<td>51.31%</td>
<td>46.15%</td>
<td>44.08%</td>
<td>31.21%</td>
</tr>
<tr>
<td>Net.(21,17)</td>
<td>77.44%</td>
<td>62.49%</td>
<td>59.23%</td>
<td>60.73%</td>
<td>58.00%</td>
<td>57.69%</td>
<td>51.32%</td>
</tr>
<tr>
<td>Net.(3,15,23)</td>
<td>85.38%</td>
<td>66.17%</td>
<td>63.37%</td>
<td>60.66%</td>
<td>62.63%</td>
<td>61.82%</td>
<td>55.61%</td>
</tr>
<tr>
<td>Net.(3,20,30)</td>
<td>83.29%</td>
<td>75.45%</td>
<td>70.90%</td>
<td>65.23%</td>
<td>62.92%</td>
<td>61.15%</td>
<td>55.12%</td>
</tr>
<tr>
<td>Net.(3,25,39)</td>
<td>80.16%</td>
<td>71.98%</td>
<td>66.62%</td>
<td>66.26%</td>
<td>63.54%</td>
<td>61.17%</td>
<td>56.08%</td>
</tr>
</tbody>
</table>

From many numerical results, it is found that: 1) the best solution of MCR-SD is often obtained from the best solution of MCR-SI; 2) the set of candidate routes $R^2_{V}$ is better listed in the order of working VP link disjoint and then working VP nondisjoint; and 3) very often, the best solutions are obtained when $\text{option}_3 = 1$ (unconditional MCR), which implies that $\text{option}_3 = 0$ may be omitted and reduce the run time by half. Further, using any of the above 16 combinations, the result is often found not far from the best solution.

The MCR heuristic is compared with formulation $F2$ and algorithm $F3 \rightarrow c$ in Table III under a single link failure scenario and with 100% restoration. The SSCA (simplest spare capacity allocation) algorithm uses the shortest routes for working VP’s and the second shortest link disjoint route for the backup VP’s. The dense networks $\text{Net}_4(15,30)$, $\text{Net}_4(20,42)$, and $\text{Net}_4(25,55)$ are subnetworks of network 4 in Fig. 2 with the average node degree $\bar{D} = 4.0, 4.2$, and 4.4, respectively. Specifically, $\text{Net}_4(15,30)$ contains nodes $0–14$ and $\text{Net}_4(20,42)$ consists of nodes $0–19$ of network 4. $\text{Net}_4(25,55)$ is network 4 itself. Similarly, the sparse networks $\text{Net}_3(15,23)$, $\text{Net}_3(20,30)$, and $\text{Net}_3(25,30)$ are subnetworks of network 3 with $\bar{D} = 3.07, 3.0$, and 3.12, respectively, where $\text{Net}_3(15,23)$ contains nodes $10–24$, $\text{Net}_3(20,30)$ consists of nodes $0–19$, and $\text{Net}_3(25,39)$ consists of nodes $0–24$ of network 3.

The results in Table III (as well as in Tables IV and V) are obtained by using route sets of APP (same as in Section IV). In Table III, the traffic demands are uniform and the number of VP’s per commodity is five ($V_\pi = 5, \pi \in \Pi$). Each VP has the same bandwidth ($=B/V_\pi$). For $F2$ and $F3 \rightarrow c$, results of both joint (j) and spare (s) optimizations are included.

As the MCR heuristic uses spare optimization, for fair comparison let us first look at algorithms SSCA, $F3 \rightarrow c(s)$, MCR, and $F2(s)$ together. It is observed that: 1) compared to $F2(s)$, the additional spare cost of SSCA is very high, more than 20% of working cost $C_{w}$; 2) MCR-SD is better than $F3 \rightarrow c(s)$, especially when the network size is large; 3) the difference in SCR between MCR-SD and $F2(s)$ is small (roughly less than 5% of $C_{w}$); and 4) the difference between MCR-SD and MCR-SI is not significant, especially in sparse networks. Note that $F2$ is for the bifurcated flow case while algorithm MCR is developed for the nonbifurcated flow situation. If converting the optimal solution of $F2(s)$ to nonbifurcated flows, the difference between MCR-SD and $F2(s)$ would be even smaller. This implies that the heuristic gives a near-optimal solution for spare capacity allocation in the nonbifurcated flow case, which is also true for nonuniform demands. The fourth observation above is interesting, which indicates that the penalty (in SCR) of using SI restoration, like the one proposed in [6], is not large, compared to the SD restoration. On the other hand, the implementation of SI restoration is relatively easier and the restoration speed is faster.

One can see from Table III that the difference between joint and spare optimizations could become very significant in large dense networks, e.g., in $\text{Net}_4(20,42)$. Moreover, algorithm $F3 \rightarrow c(j)$ is better than the MCR algorithm in dense networks, but it is not always true in large sparse networks. This observation suggests to use algorithm $F3 \rightarrow c(j)$ in large dense networks whenever possible. The impact of the number of VP’s per commodity $V_\pi$ on the SCR is illustrated in Table IV, using dense network $\text{Net}_4(20,42)$ and sparse network $\text{Net}_3(20,30)$ as examples. It is found that the SCR is not very sensitive to $V_\pi$ which seems to confirm the results in Section IV-D on LP versus IP. Hence, the initial conclusions drawn from Table III still hold even if each commodity contains only one VP. Further, increasing $V_\pi$ does not always lead to the decrease in SCR (see $V_\pi = 10$ in Table IV), which is due to the heuristic nature. Similar results are obtained under nonuniform traffic demands as shown in Table V, where each node pair has only one VP ($V_\pi = 1$).

VI. SURVIVABILITY IN EXISTING NETWORKS

In the previous two sections, we have dealt with the optimal/near-optimal network design problems. In this section,
we investigate the survivability (restoration ratio) in existing networks. The purpose is to compare network survivability for link and path restorations.

In an existing network, the arc capacity $c_a$, $a \in A$ and real traffic flows (i.e., working VP’s) are known. The spare capacity $c_a^{\text{sparse}}$ on arc $a$ is just equal to the remaining unused capacity on that arc. Failure-oriented reconfiguration is considered here. Again, $x_{\pi r}^s$ is used to represent a (normalized) traffic flow of commodity $\pi$ on route $r$, $r \in R^s_0$, $\pi \in \Pi$, and $y_{\pi r}^s$ denotes a restoration flow on route $r$ for the affected commodity $\pi$ when the network is in failure state $s$, $r \in R^s_0$, $\pi \in \Pi$, $s \in S - S_0$. Similar to $F_2$, it is readily seen that for the path restoration the network survivability can be formulated as the following LP problem.

**F4:**

maximize

$$
\sum_{s \in S - S_0} \sum_{\pi \in \Pi} \sum_{r \in R^s_0} y_{\pi r}^s
$$

subject to

$$
\sum_{r \in R^s_0} \delta_{\pi r} x_{\pi r}^s y_{\pi r}^s \leq c_a^{\text{sparse}}, \quad a \in A, \quad s \in S - S_0 \quad (15)
$$

$$
\sum_{r \in R^s_0} y_{\pi r}^s - \sum_{r \in R^s_0} \Omega(\pi, r, s) x_{\pi r}^s \leq 0, \quad \pi \in \Pi, \quad s \in S - S_0, \quad (16)
$$

In the above formulation, constraints (15) guarantees that the capacity used by the restoration flows passing each arc will not exceed the available spare capacity on that arc. Constraints (16) ensures that the volume of restoration flow will not larger than the affected traffic flows for every commodity. Similar to $F_3$, one can also easily obtain the network survivability formulation for the link restoration.

Since the network can only be in one of the failure states at a time, the above formulation will give the maximal amount of restored traffic, denoted by $t_0(s)$, for a given network failure state $s$, $s \in S - S_0$. The restoration ratio $\theta(s)$ in state $s$ is equal to the ratio of the restored traffic $t_0(s)$ to the (restorable) affected traffic $t_a(s)$, i.e., $\theta(s) = t_0(s)/t_a(s)$. The network survivability is defined here as the average restoration ratio over all possible failure cases

$$
\eta = \frac{\sum_{s \in S - S_0} \theta(s)}{|S - S_0|}.
$$

Another possible way of defining the network survivability is to use the aggregate restoration ratio given by

$$
\eta^* = \sum_{s \in S - S_0} t_0(s)/\sum_{s \in S - S_0} t_a(s).
$$

For a given SCR or equivalently spare cost $C_s$ [see (13)], how the spare capacity is distributed among the network arcs when a failure occurs will depend on the original capacity allocation and the current traffic pattern. There may be numerous possibilities. In our study, four different ways of distributing the spare capacity are considered:

1) each arc has the same spare capacity, i.e., $c_a^{\text{sparse}} = C_s/\sum_{a \in A} d_a$;
2) each arc has the same spare cost, i.e., $c_a^{\text{sparse}} = C_s/(|A| d_a)$;
3) the spare capacity on each arc is proportional to the working capacity on that arc, i.e., $c_a^{\text{sparse}} = c_{a\text{work}} \cdot SCR$;
4) the spare capacity on each arc is inversely proportional to the working capacity on that arc, which yields $c_a^{\text{sparse}} = C_s/(c_{a\text{work}} \cdot |A| d_a/c_{a\text{work}})$. SCD$_{n4}$ is used to represent $n$th spare capacity distribution, $1 \leq n \leq 4$.

Assuming uniform traffic demands and SCD$_{1}$ (i.e., each arc has the same spare capacity), the average restoration ratio $\eta$ (also the aggregate restoration ratio $\eta^*$) in Net$_3(20, 30)$ is depicted in Fig. 9 as a function of SCR, under a single link failure scenario. As before, all working VP’s take the shortest paths and the path set of APP is used in obtaining the results. Fig. 9 illustrates that $\eta$ grows rapidly when the SCR increases from 0% to 50% and the growth slows down when the SCR further increases. Moreover, for a given SCR the restoration ratio is larger when using the path restoration and the difference is more pronounced if using $\eta^*$. Note that $\eta^* < \eta$. If the spare capacity is optimally distributed, using the path restoration, SCR = 61.15% is sufficient to ensure 100% restoration under a single failure scenario as indicated in Fig. 9.

Similar results are observed for different spare capacity distributions, various network topologies, and nonuniform...
traffic demands. Some numerical examples are shown in Tables VI and VII under uniform and nonuniform traffic demands, respectively. It appears that evenly distributing spare capacity among network links often gives higher restoration ratio for the tested network examples and traffic patterns. Similar observations are also made for $\eta^*$. 

### VII. CONCLUDING REMARKS

This paper has compared several restoration strategies for the self-healing ATM networks quantitatively in terms of SCR. It is found that path restoration could lead to significant savings in SCR over link restoration, especially in large sparse networks. In path restorations, the additional spare cost of failure-oriented reconfiguration is not very large compared to global reconfiguration. Interestingly, this additional cost could be greatly reduced by reusing the capacity occupied by the failed working VP’s. Further, in many cases, the difference in SCR between SI and SD restorations is not very significant when using spare optimization. In those cases, the SI restoration appears to be attractive because its implementation is easier. However, in some cases, the SCR difference between SI and SD restorations cannot be ignored, e.g., when using joint optimization. In these cases, the choice of which restoration strategy may depend on other factors, like restoration speed, VPI redundancy, nodal processing/memory requirement, etc., which remain for further study. It is also found that treating traffic demand between a node pair as an integral (one super VP) might not significantly increase the SCR but have potential to speed up the restoration process, which also remains for further investigation.

Two new heuristic algorithms called $F3 - c$ and MCR are presented in the paper for the design of large self-healing ATM networks using path restoration. The $F3 - c$ algorithm is based on LP techniques and can perform both joint and spare optimizations. The MCR algorithm is developed for the nonbifurcated flow case and can be applied to both SD and SI restorations. Numerical results show that the MCR algorithm is efficient and able to provide near-optimal solutions for spare capacity allocation. Currently we are extending the MCR algorithm to cope with other design issues, like capacity modularization and nonlinearity of VP bandwidth and link capacity. Note that the MCR heuristic and the IP version of formulations $F1$ and $F2$ could also be used for self-healing STM networks by restricting the bandwidth of each VP to STM-$\eta$ (or OC-$\eta$).

### TABLE VII

<table>
<thead>
<tr>
<th>Ratio</th>
<th>SCR</th>
<th>link</th>
<th>0.3</th>
<th>0.5</th>
<th>1.0</th>
<th>0.3</th>
<th>0.5</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net.3[20,30]</td>
<td>SCD.1</td>
<td>56.71%</td>
<td>77.39%</td>
<td>95.67%</td>
<td>11.99%</td>
<td>10.64%</td>
<td>4.07%</td>
<td></td>
</tr>
<tr>
<td>SCD.4</td>
<td>47.89%</td>
<td>68.59%</td>
<td>88.25%</td>
<td>12.72%</td>
<td>10.67%</td>
<td>7.13%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net.4[20,42]</td>
<td>SCD.1</td>
<td>73.07%</td>
<td>92.64%</td>
<td>99.53%</td>
<td>11.26%</td>
<td>3.29%</td>
<td>0.07%</td>
<td></td>
</tr>
<tr>
<td>SCD.4</td>
<td>59.90%</td>
<td>79.16%</td>
<td>98.33%</td>
<td>18.06%</td>
<td>14.75%</td>
<td>1.40%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ACKNOWLEDGMENT

The authors wish to thank the reviewers for their valuable comments and suggestions in improving the paper.

### REFERENCES


Yijun Xiong (M’93) received the B.S. and M.S. degrees from Shanghai Jiao Tong University, China, in 1984 and 1987, respectively, and the Ph.D. degree from the University of Ghent, Belgium, in 1994, all in electrical engineering. Since 1996, he has been an Assistant Professor of Electrical and Computer Engineering at Louisiana State University, Baton Rouge. From 1989 to 1992, he was with the Research Center of Alcatel Bell Telephone, Antwerp, Belgium, from 1992 to 1994. He was with the Laboratory for Communications Engineering, University of Ghent, Belgium, and from 1994 to 1996 he was with INRS-Telecommunications, Montreal, Canada. His current research interests include multicast ATM switches, network resource management, network survivability, and queueing theory.

Lorne G. Mason received the B.Sc. and Ph.D. degrees in mechanical engineering from the University of Saskatchewan, Canada, in 1963 and 1972, respectively.

He is currently an Emeritus Professor at INRS-Telecommunications, University of Quebec, and a Consultant in Telecommunications Research. From 1963 to 1965, he was with Bristol Aerojet in Winnipeg, MB, Canada, where he was involved in the design of the black brant rockets. He joined the British Columbia Telephone Company as a Traffic Engineer in 1966 and again in 1972 as a Consultant for planning digital networks. In 1973, he served as a Consultant to Yale University, where he collaborated with Prof. K. S. Narendra in pioneering the use of learning automata for adaptive routing in telecommunication networks. During 1974–1977, he was with Bell-Northern Research, where he developed planning tools and methods for digital network evolution and state-dependent routing. Since 1977, he has been with INRS-Telecommunications, where he was an Invited, Associate, and Full Professor, and recently Professeur Honoraire. During this period, he conducted research and taught post-graduate students in telecommunications. He is also a Professeur Associe’e at ENST (Telecom Paris). His primary research interests are the application of control theory and operations research methods to telecommunication network design, control, and management. He has held numerous industrial research contracts and strategic grants from the National Scientific and Engineering Research Council of Canada in the area of broadband network design and analysis and has more than 50 publications on the subject in leading conferences and journals.

Dr. Mason is a participant in the Network of Centers of Excellence program, sponsored by the Canadian Government, where he is Project Leader in the area of broadband network control. He was co-recipient of the 1993 Stentor Award for collaborative research in telecommunications for his contributions to state-dependent routing. He organized and chaired the joint IFIP/IEEE Conference, Broadband Communications’96, and was the Editor of the Proceedings. He was the Technical Co-Chairman for the IEEE ATM’98 Workshop, and a Technical Program Committee Member for several international conferences in the broadband area. He is a member of the Canadian Institute for Information Processing (CIPS) member and a Canadian representative to the Telecommunications Committee of the International Federation for Information Processing (IFIP TC6).