8

A comparison of pre-planned routing techniques for virtual path restoration

P.A. Veitch and D.G. Smith
Dept. of Electronic & Electrical Engineering,
University of Strathclyde,
Glasgow G1 1XW,
Scotland, U.K.
Tel: (0141) 552 4400
Fax: (0141) 552 4968
E-mail: {pveitch,g.smith}@comms.eee.strathclyde.ac.uk

I. Hawker
BT Laboratories,
Martlesham Heath,
Ipswich IP5 7RE,
England, U.K.

Abstract

Network restoration techniques will be vital to ensure B-ISDN service survivability in the event of high capacity link and node failures. Reliable ATM crossconnect networks can be implemented by the strategic pre-assignment of protection Virtual Path (VP) routes to permit recovery from a realistic subset of all possible failures, eg single span failures. The method of protection route assignment influences the quantity of redundant resources like spare capacity and Virtual Path Identifiers (VPIs), whilst nodal hardware costs are incurred due to the requirement of pre-stored alternate routing information. In addition to implementation costs, the impact that the choice of rerouting scheme has on other factors must be considered. For example, the degree of path elongation following restoration may adversely affect the delay performance of certain connections. Also, the amount of computation required to design the protection routes, and the effort needed to activate such routes have to be taken into account. This paper formulates metrics to facilitate a comparative evaluation of four distinct routing strategies for VP restoration, and in conjunction with a discussion of qualitative properties of each scheme, it concludes that failure independent rerouting is the preferred approach.

Keywords

ATM Virtual Paths, restoration, routing, survivable network design

D. D. Kouvatsos (ed.), *ATM Networks*

1 INTRODUCTION

Because the potential repercussions of a cable break or node failure in a high capacity broadband trunk network are so great, survivability is crucial (Wu, 1992). Restoration is the process of re-establishing trunk groups affected by a failure by exploiting spare capacity at diverse locations in a mesh topology (Veitch et al, 1995b). This is realised by high speed Digital Crossconnect Systems (DCSs) which are managed centrally, but also have the capability to interact in a distributed fashion, enabling fast restoration. If restoration is rapid enough, active calls may not be dropped. Indeed, a target completion time of 2 seconds would ensure preservation of the majority of voice connections (Sosnosky, 1994). Recent research into ATM Virtual Path (VP) restoration suggests that progress can be made in achieving very fast service recovery (Kawamura et al, 1994, Anderson et al, 1994, Veitch et al, 1995c). This is largely attributed to the logical nature of a Virtual Path which decouples routing and capacity assignment making reconfiguration simple compared with Synchronous Transfer Mode (STM) paths (Sato et al, 1990). This paper focuses on VP restoration in ATM networks, and in particular, the range of approaches to pre-planning alternate routes for this purpose.

From a network operator's point of view, a restoration strategy should be simple to implement, and resource efficient. In tandem with these requirements, the scheme should offer the subscriber fast service recovery from a wide range of failures. A suitable approach therefore, is to pre-assign restoration paths in advance of failure occurrence. This can be performed by a centralised computer with a global view of the network; resources can be managed efficiently and an appropriate subset of failures can be selected as the basis for protection. In the event of a failure, distributed signalling between crossconnects can be used to achieve very fast restoration with a simple protocol since it is only necessary to activate pre-determined routes. Two distinct methods of establishing protection VP routes which have been identified in the literature are categorised as failure dependent (Anderson et al, 1994) and failure independent (Kawamura et al, 1994) rerouting, both of which are defined later. Although both of these techniques constitute pre-assigned VP restoration, they are fundamentally different in certain aspects of implementation and performance, hence it is vital to perform a formal comparison. We focus on single span failure which is the simultaneous failure of all the transmission systems between two crossconnect nodes. This assumption facilitates a fair comparison of schemes, since the description of one of the two paradigms studied accounts for span failures only (Anderson et al, 1994).

The costs of implementing a particular restoration system are affected by the required spare resources such as link/buffer capacity and Virtual Path Identifiers (VPIs), as well as the memory overheads required to support the pre-storage of alternate routing information. In addition, different rerouting techniques can be assessed in terms of the computational effort required to design the protection plans, as well as the signalling effort needed to activate protection routes. With respect to performance, the choice of rerouting strategy affects the user-perceived quality of service, since restoration often induces path elongation, causing increased delays and cell delay variation. Following a simple description of the alternative rerouting schemes in section 2, a comparative evaluation will be carried out in section 3 using metrics based on required spare capacity, VPI redundancy, path length elongation, storage overheads and the computational effort ascribed to protection route design. Section 4 discusses other qualitative factors that can be employed to compare the two distinct rerouting paradigms, including the signalling protocol and robustness in the presence of uncertainty. Section 5 concludes the paper by reasoning in favour of one particular method by taking into account all the quantified metrics of section 3 as well as the implementation aspects considered in section 4.

2 ALTERNATE VP REROUTING SCHEMES

Prior to providing a comparative evaluation, three failure dependent rerouting policies will be described, followed by an overview of the failure independent rerouting algorithm. Throughout, it is assumed that the working VP configuration is known a priori and that single span failure protection is required.

2.1 Failure dependent approaches

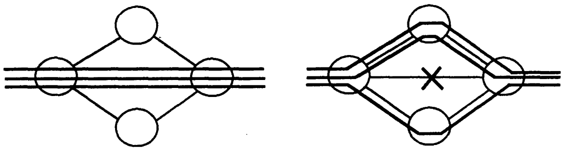
With failure dependent rerouting, each possible single span failure is examined in turn, and alternative routes are subsequently found for all the VPs affected by the failure. A batch alternate route planning operation of this kind may be written:

```
For each possible span failure
    For each failed VP
        Find alternate path according to rerouting policy
    End For
End For
```

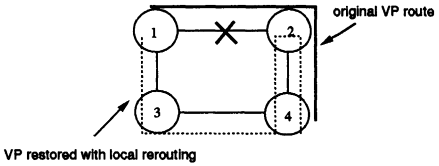
Hence, there is a unique reconfiguration associated with each failure. Alternate routing data in the form of VPI and link ID information are stored in databases of all the relevant crossconnect nodes. When a span fails, the ID of the failed span is broadcast to all network nodes which re-load their lookup tables with the relevant data, resulting in an asynchronous logical (i.e. VP) topology update (Anderson et al, 1994). Three separate versions of failure dependent rerouting will now be explained.

2.1.1 Local rerouting

In this scheme, when a span fails, all the affected VPs are rerouted between the terminating nodes of the span, without regard to the source and destination of the VPs (Figure 1 (a)).



(a) Local rerouting



(b) backhauling

Figure 1: Characteristics of local rerouting

Although extremely simple to compute the alternate routes, and execution of restoration is potentially fast due to the majority of message processing being carried out in the vicinity of the failure, this is a greedy algorithm and can cause the undesirable phenomenon known as *backhauling* (Figure 1 (b)). From the figure, failure of span 1-2 leads to an alternate route being computed between nodes 1 and 2 as 1-3-4-2. The failed path 1-2-4 consequently uses the route 1-3-4-2-4 meaning span 2-4 is utilised twice.

2.1.2 Local-destination rerouting

A potentially more efficient result should be possible with a more sophisticated algorithm, such as "local-destination" rerouting proposed by AT&T (Anderson et al, 1994). Considering a span failure, failed VPs will be rerouted with one of the span terminating nodes as the starting point. The destination of the alternate route will depend on the individual VP route however, so as to reduce resource consumption. In Figure 2, if span 3-4 fails, then devising a shortest hop path between node 3 and the VP terminating node 8 will produce the two hop detour 3-7-8; a more efficient result than *pure* local rerouting. The essence of the algorithm is to retain as large a portion of the original working path route as possible, then find the most direct path to the destination of the failed VP whilst avoiding the failed span. From this very basic analysis therefore, a set of heuristics can be devised with respect to an individual VP affected by failure of a span:

1. The starting point of the detour is the terminating node of the failed span at the side of the VP with most hops: if equal, select at random.
2. Find the shortest hop path between starting point of detour and VP destination node.

3. Add retained part of path.
4. Remove self-loops and discount overlapping resource demands.

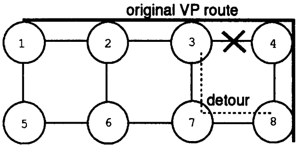


Figure 2: Characteristics of local-destination rerouting

Step 4 is included because backhauling is still possible, as can be seen in Figure 3 by the failure of span 2-3. The path is of equal length on each side of the failed span (1 hop). The starting point of the detour is selected as node 3, so we retain hop 6-3 of the original VP route, and seek a shortest hop path between node 3 and the VP termination, which is node 1. The result of the detour is thus 3-6-5-2-1, and when we concatenate this with the retained part of the original path, we obtain 6-3-6-5-2-1. Obviously, backhauling has occurred due to the self-loop 6-3-6, so this is eliminated leaving 6-5-2-1 as the new VP route employed due to failure of span 2-3. A final check to be made is whether or not the alternate route uses any of the original VP route hops; this occurs in the example with respect to span 1-2, hence the spare capacity/VPI requirements for this hop are discounted.

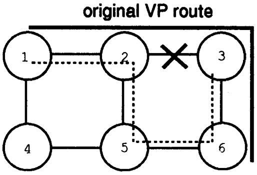


Figure 3: Self-loop and overlap during local-destination rerouting

2.1.3 Source-based rerouting

Source-based rerouting ought to be yet more efficient in terms of spare capacity (Anderson et al, 1994), by allowing alternate routes to be computed between the terminating nodes for each path affected by a failed span (Figure 4). The effect of this algorithm is to spread the demand for spare capacity more freely throughout the network than the preceding two schemes described. Any overlap between the original path and designated alternate route (eg span 1-2 of Figure 4) is dealt with by discounting the spare resource demands for such spans.

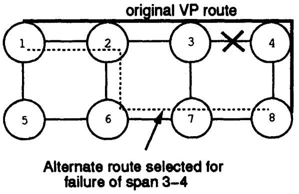


Figure 4: Source-based rerouting

2.2 Failure independent approach

In the failure independent case, a single alternate VP route can be designed to protect a working VP from any single span failure. The design criterion to satisfy this requirement is that a span disjoint route be selected for protection. Regardless of the underpinning physical span which induces VP failure therefore, the same protection route is employed for restoration. From Figure 5, whether span 1-2, 2-3 or 3-4 fails, the same backup route, 1-5-6-7-8-4 protects the working path 1-2-3-4. Indeed, failure of nodes 2 or 3 may be circumvented by activating this same route. Because there is a single alternate protection route for a working VP, the backup can be established in advance of failure by setting VPIs at the appropriate crossconnect nodes; from Figure 5, this corresponds to nodes 5, 6, 7 and 8. Activation of such a VP may be performed by altering the routing table at the connection endpoints (i.e. nodes 1 and 4 from the Figure). It is at such nodes that storage of alternate routing data is required.

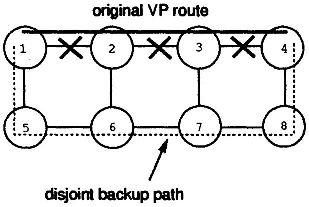


Figure 5: Failure independent (span disjoint) rerouting

The algorithm may be written:

```
For each VP
    Find shortest hop span disjoint path
End For
```

At the heart of this, and all of the failure dependent strategies described previously, is a shortest path computation. The common algorithm employed is the Floyd-Warshall technique, based on distance matrix manipulation. A very simple modification is made in that given the choice of two equidistant path routes, a random selection between the two is made. It should be reiterated that for all the rerouting schemes investigated, shortest hop paths are found based on link weights of unity. On a batch provisioning basis, this produces suboptimal results in terms of spare capacity. If minimisation of the global spare capacity is required, a more complex solution to the rerouting problem would be required, such as mathematical programming or stochastic techniques based on simulated annealing (Coan et al, 1991). The techniques employed for the comparative evaluation detailed in the following section are not optimised in terms of spare capacity requirements so as to ensure that none of the other metrics which are quantified become negatively biased.

3 COMPARATIVE EVALUATION

3.1 Network assumptions

A variety of network models will be used to generate performance data for each of the four VP rerouting methods. Some pre-requisites are essential to simplify the analysis. The networks are meshed backbones comprising VP crossconnects, each of which is assumed to be collocated with a VC switch. Measurements of required resources (spare capacity and VPIs) and path lengths correspond to the inter crossconnect spans, not the links between VC and VP switching elements. Each span between crossconnect nodes will carry just one bidirectional fibre transmission system, enabling simple computation of redundant resources. In each network considered, a single bidirectional VP of unit capacity is established between each node pair, hence in an n node network, there are $n(n-1)/2$ Virtual Paths. These working paths will be generated using the Floyd-Warshall algorithm with shortest hop routes being selected. Alternate routing information which forms the basis of the design metrics for comparison, is then produced for each of the four schemes detailed in the previous section. Full protection from single span failures is provided in each case.

Prior to the analytic detail of individual metrics, some basic nomenclature is introduced. The physical network is described as a graph $G(V,E)$, whereby V is the set of vertices representative of ATM nodes, and E is the set of edges representing inter-nodal spans. A single vertex is denoted v ($v \in V$) whilst a single edge is symbolised as e ($e \in E$). The working capacity of an edge e is denoted W_e, whilst the spare capacity is S_e. The logical network is described by the set of paths P, whereby a single path π ($\pi \in P$) is the collection of edges traversed. The capacity of a path π is C_π. The set of protection routes is defined as $\hat{P}$, with a protection path pertaining to a working path π denoted $\hat{\pi}$ in the failure independent case and $\hat{\pi}^{\mathrm{f}}$ in the failure dependent case, with the superscript f denoting the index of the failed edge, e_{f}. Note that $\hat{\pi}^{\mathrm{f}}$ represents the end-to-end route of the path π following restoration from failure of edge e_{f}, part of which is often unchanged. For clarity, we further define $\hat{\pi}^{\mathrm{f}}_{\mathrm{d}}$ to be the edges of the rerouted part of the path only (the subscript d refers to *detour*). There are m edges, n vertices and k paths in the network. Additional notation will be introduced where necessary.

3.2 Computation of metrics

Given the working VP and alternate routing information, the following metrics can be computed for each of the four rerouting schemes applied to several network topologies.

3.2.1 Spare Capacity Ratio (SCR)

Ultimately, the SCR is the ratio of the aggregate spare capacity in the network to the aggregate working capacity. The working capacity of an edge is found by summing the capacities of constituent paths:

$$W_{\mathrm{e}} = \sum_{\pi \in P, e \in \pi} C_{\pi}. \tag{1}$$

Hence, the total working capacity is found by summing (1) over the set of network edges. Computing the individual spare capacity quotas per edge is a little more complex. Depending on the edge which has failed in the network, the demanded spare capacity on the remaining edges differs. This is because a different set of working paths will be affected by each possible failure, hence a different reconfiguration is performed in each case. Letting $S_{\mathrm{e}}^{\mathrm{f}}$ symbolise the spare capacity required on edge e due to failure of edge e_{f}, we have:

$$S_{\mathrm{e}}^{\mathrm{f}} = \sum_{\pi \in P, e_{\mathrm{f}} \in \pi, e \in \hat{\pi}} C_{\pi}. \tag{2}$$

For the failure independent case, or:

$$S_{\mathrm{e}}^{\mathrm{f}} = \sum_{\pi \in P, e_{\mathrm{f}} \in \pi, e \in \hat{\pi}_{\mathrm{d}}^{\mathrm{f}}} C_{\pi}. \tag{3}$$

Which applies to the failure dependent case. Now, the provisioning of spare capacity on each edge must account for the edge failure which will yield the greatest demand for rerouted traffic. We thus find the required spare capacity for an edge e, denoted S_{e}, in the following way:

$$S_{\mathrm{e}} = \max\{S_{\mathrm{e}}^{1}, S_{\mathrm{e}}^{2}, ..., S_{\mathrm{e}}^{\mathrm{m}}\}. \tag{4}$$

It should be stressed that no attempt is made at capacity modularisation so as to conform to specific transmission systems. The value of SCR is subsequently found by dividing the total spare capacity by the total working capacity:

$$SCR = \frac{\sum_{e \in E} S_{\mathrm{e}}}{\sum_{e \in E} W_{\mathrm{e}}}. \tag{5}$$

3.2.2 Mean VPI Redundancy (*MVR*)

When protection routes are designed, Virtual Path Identifiers (VPIs) must be reserved for the appropriate links. The total number of idle VPIs is a function of the number of edges used in each protection route since VPI translation is performed for each link of a VP connection (ITU-T, 1993a). For the failure independent case, letting $L(\hat{\pi})$ be the length (number of edges used) of a specific protection path, $\hat{\pi}$, the total number of idle VPIs, denoted N^{v}, is found from:

$$N^{\mathrm{v}}_{\mathrm{fi}} = \sum_{\hat{\pi}\in\hat{P}} L(\hat{\pi}). \tag{6}$$

The subscript fi indicates *failure independent.* In a similar fashion, fd will specify the *failure dependent* version of appropriate metrics. For the failure dependent case, $L(\hat{\pi}^{\mathrm{f}}_{\mathrm{d}})$ is the number of spans in the rerouted part of the end-to-end working path π, activated due to failure of e_{f}. Considering all failures per path, and the complete set of paths in the network:

$$N^{\mathrm{v}}_{\mathrm{fd}} = \sum_{\pi\in P}\sum_{e_{\mathrm{f}}\in\pi} L(\hat{\pi}^{\mathrm{f}}_{\mathrm{d}}). \tag{7}$$

Now, given the total number of VPIs, regardless of the rerouting scheme, the MVR may be found by dividing the appropriate N^{v} by m (the number of edges) giving the mean VPI redundancy per edge; this quantity can then be normalised to the maximum number of VPIs per link (4096), yielding:

$$MVR = \frac{N^{\mathrm{v}}/m}{4096}. \tag{8}$$

3.2.3 Path Elongation Factor (*PEF*)

This is simply the ratio of the mean length of a VP rerouted during failure restoration to the mean working path length. For the failure independent scheme, this is given by the equation:

$$PEF_{\mathrm{fi}} = \frac{\sum_{\hat{\pi}\in\hat{P}} L(\hat{\pi})}{\sum_{\pi\in P} L(\pi)}. \tag{9}$$

For the failure dependent scheme, we first find the mean rerouted path length for an individual path, denoted $\bar{n}^{\mathrm{r}}_{\pi}$, given by:

$$\bar{n}^{\mathrm{r}}_{\pi} = \frac{\sum_{e_{\mathrm{f}}\in\pi} L(\hat{\pi}^{\mathrm{f}})}{L(\pi)}. \tag{10}$$

Now, averaging over all paths in the network provides the mean rerouted path length, which, when divided by the mean working path length gives the *PEF* for the failure dependent rerouting as:

$$PEF_{\mathrm{fd}} = \frac{\sum_{\pi \in P} \bar{n}_{\pi}^{\mathrm{r}}}{\sum_{\pi \in P} L(\pi)}. \tag{11}$$

3.2.4 Mean Memory Requirements (*MMR*)

The memory requirements for pre-stored data are quite distinct for each approach to rerouting. With the failure dependent technique, VPIs and link IDs are associated with specific failures. The information is stored in a database and is only loaded into the active VP routing tables when the crossconnect is notified of the failure. Such an operation is carried out at all the participating nodes of an alternate route detour. In contrast to this, the failure independent approach involves pre-loading translation tables of all downstream nodes with the VPIs of the alternate route; at such nodes, there are *no* database memory requirements for VPI/link ID information. This is because the translation table itself contains the mapping between input and output VPIs. Since such tables will be designed for the maximum possible number of VPs passing through a node, there is effectively no overhead. It need only be at the VP endpoints that alternate VP routing data be stored at a database, which is used to re-load the translation tables when these nodes learn that the VP has failed (Veitch et al, 1995a). Some simple assumptions will now be made to enable an approximate enumeration of memory requirements for the two rerouting paradigms. For the failure independent method, an alternate (VPI(out)/Link(out),VPI(in)/Link(in)) pairing is associated with a bidirectional VP at each endpoint, as depicted in Figure 6(a).

Working VP	**Alternate VP**	
VPI(out) Link(out)	VPI(out) Link(out)	downstream
VPI(in) Link(in)	VPI(in) Link(in)	upstream

(a)

Span ID	VPI(in) Link(in)	VPI(out) Link(out)

(b)

Figure 6: Storage format for alternate routing information: (a) failure independent (b) failure dependent

This covers the upstream and downstream parts of the VP. We assume that each entry takes 16 (2 × 8) bytes of memory for storage. If one bidirectional span disjoint protection path is allocated to each of k bidirectional VPs in a network, the total memory requirement is simply $2 \times k \times 16$ bytes. Thus, the *MMR* metric which gives the average memory requirement per node is simply:

$$MMR_{\mathrm{fi}} = \frac{32 \cdot k}{n}. \tag{12}$$

For failure dependent techniques, the input and output VP information corresponding to a particular span failure for one direction of a VP, is shown in Figure 6(b). It is assumed that 10 bytes are consumed with this format. If $L(\hat{\pi}_{\mathrm{d}}^{\mathrm{f}})$ span hops are used in a certain protection detour, information storage is required at $L(\hat{\pi}_{\mathrm{d}}^{\mathrm{f}})+1$ nodes. Thus, in any failure dependent scheme, the memory required for a bidirectional alternate route employed when a specific span fails is:

$$2 \times (L(\hat{\pi}_{\mathrm{d}}^{\mathrm{f}}) + 1) \times 10 \quad \mathit{bytes}.$$

To compute the total memory requirements for a network, the above quantity will be summed over all possible span failures related to all bidirectional VPs. The *MMR* is then found by dividing by n, the number of nodes, to give:

$$MMR_{\mathrm{fd}} = \frac{20 \times (\sum_{\pi \in P} \sum_{e_f \in \pi} (L(\hat{\pi}_{\mathrm{d}}^{\mathrm{f}}) + 1))}{n}. \qquad (13)$$

3.2.5 Routing Computational Effort (RCE)

The computation required to produce alternate routes is non-trivial since working VP configurations may be subject to capacity and/or routing re-allocation (Sato et al, 1990) at regular intervals. This implies that protection plans have to be revised in accordance with the new VP arrangement. Fast computation is thus essential to minimise the probability that a failure will occur between the time of the working VP rearrangement and the assignment of new protection routes. We express the RCE metric in the simplest possible way, that is by the number of rerouting computations for the required protection condition, assumed throughout to be single span failures. For the failure independent scheme, since there is a protection path for each of the k working paths, we have:

$$RCE_{\mathrm{fi}} = k. \qquad (14)$$

For the failure dependent scheme, alternate routes are found for each failed path of every failed span. The total number of alternate routes required can hence be found by summing the number of spans used in each path to obtain:

$$RCE_{\mathrm{fd}} = \sum_{\pi \in P} L(\pi). \qquad (15)$$

3.3 Numerical results

A computer program was written which takes any network topology description as its input, and produces the above metrics as its output by realising each of the alternate routing strategies. Shortest path routes were found for working paths with a random choice between equal length paths. For simplicity, all working VPs were assumed to be of unit capacity. The SCR, MVR, PEF, MMR and RCE metrics were computed for four grid networks of 6, 9, 12 and 20 nodes. Because of the random outcome of the shortest path algorithm, the mean result from 5 replicated computations was derived. Figures 7 and 8 display the SCR and MVR results, respectively. In all graphs, plotted points are joined up for visual convenience.

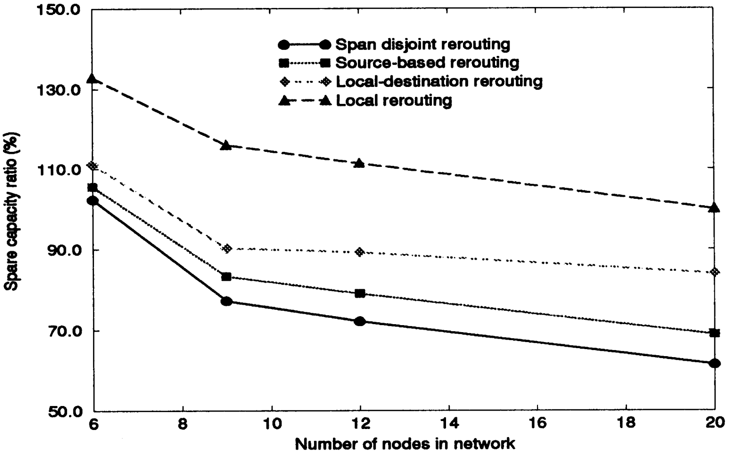


Figure 7: Spare Capacity Ratio (SCR) versus network size

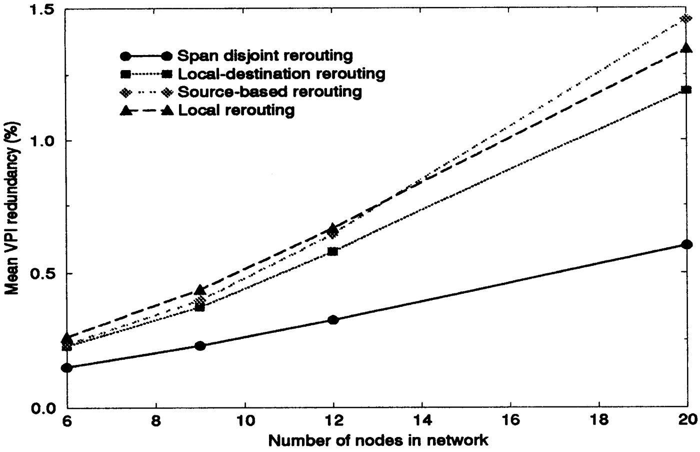


Figure 8: Mean VPI Redundancy (MVR) versus network size

In Figure 7, it can be seen that the failure independent i.e. span disjoint, rerouting scheme produces the lowest spare capacity ratio. As to why this is better than the source-based rerouting, it could be argued that by adopting the constraint of disjointness, backup routes are *forced* to spread the demand for spare capacity around the network. In the source-based rerouting meanwhile, the protection routes may often re-use original VP links, thus concentrating spare capacity requirements on links closer to the failure as in the local-destination approach. Of the three failure dependent approaches meanwhile, the source-based rerouting method requires the least spare capacity than the others which is thanks to the greater degree of freedom in route selection. The local-destination rerouting improves the efficiency of alternate routing design over local rerouting due to the elimination of backhauling.

From Figure 8, the VPI redundancy is greater for the failure dependent approaches, and the divergence between these and the failure independent scheme increases with network size. The main reason for this is that because different routes are allocated to individual failures which may affect a given VP, many more links are potentially involved in the rerouting process. Although the failure independent scheme used less VPIs than all of the failure dependent methods in the examples considered, this need not always be the case. One of the reasons that less than 100% spare capacity is needed for failure protection in a mesh network, is that sharing of resources between possible failures (equation (4)) is exploited. This sharing of resources between disparate failure events may be applied to VPIs. In the computations so far, a different VPI is employed for each span of every protection route, regardless of whether or not they correspond to different failures. As with capacity sharing however, the same VPI may be re-used across different failures. This is feasible in the failure dependent rerouting schemes since VPIs are stored in databases, only to be loaded into lookup tables upon failure notification. The prospect of "VPI sharing" presents an advantage of failure dependent over failure independent rerouting. This is because it is not feasible to have VPIs shared amongst protection paths defined by active VPI entries in lookup tables, since ambiguous routing would accrue.

We revise the *MVR* for the failure dependent case by defining an integer I_e^f to be the number of VPIs needed on edge e due to failure of edge e_f. The worst-case quantity of VPIs required on an edge e, is thus:

$$I_e = \max\{I_e^1, I_e^2, ..., I_e^m\} \tag{16}$$

Hence, the total number of reserved VPIs will be:

$$N_{fd}^{v} = \sum_{e \in E} I_e \tag{17}$$

Which can be used in equation (8) to provide the *MVR*. The *MVR* metric was subsequently recomputed with VPI sharing allowed in the failure dependent schemes, and as shown in Figure 9, the result is a lower mean redundancy of VPIs than the span disjoint rerouting technique. Where VPI numbers are re-used between different failures, the relative order of failure dependent schemes in terms of increasing resource demand is the same as that for the SCR metric.

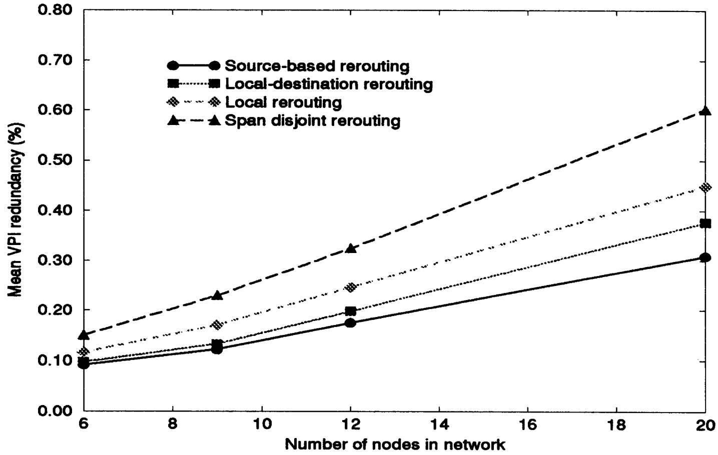


Figure 9: Recomputed *MVR* with VPI sharing, versus network size

The *PEF* metric is shown in Figure 10.

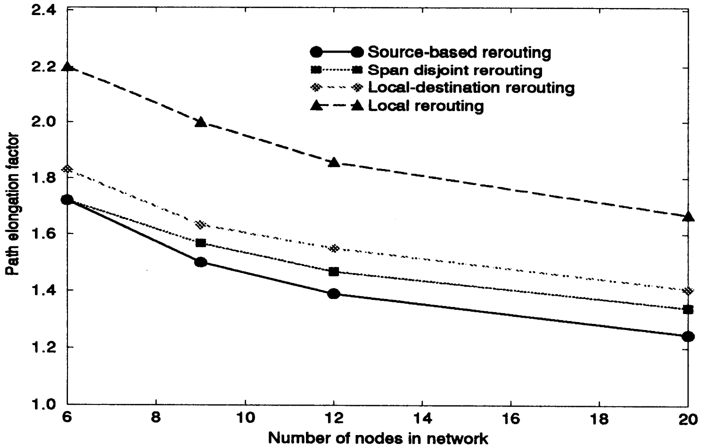


Figure 10: Path Elongation Factor (*PEF*) versus network size

From Figure 10, the minimal path elongation effects are evident with source-based rerouting, improving over the span disjoint rerouting results. The local-destination is sizeably better than local rerouting with the latter demonstrating greatest sensitivity to path elongation, mainly due to backhauling effects. To help understand why source-based rerouting should outperform span disjoint rerouting when both techniques reroute from path terminating nodes, consider Figure 11.

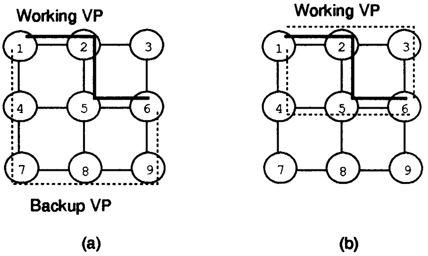


Figure 11: Potential path elongation with span disjoint rerouting

In part (a) of the Figure, the working path 1-2-5-6 is shown to be protected by backup route 1-4-7-8-9 with the failure independent rerouting scheme. There is thus a difference of 2 hops between working and protection routes. Referring to part (b) of the diagram, with the source-based rerouting version of failure dependent protection, route 1-4-5-6 could be selected for the failure of spans 1-2 or 2-5. For the failure of span 5-6 meanwhile, the new route could be 1-2-3-6. In all such cases, the working and protection routes are the same length, i.e. there is no elongation. The inferior PEF of failure independent rerouting is thus due to the disjoint criterion. It should be pointed out however, that with a more careful selection of working path route between the same nodes in Figure 11, eg 1-2-3-6, a span disjoint backup path 1-4-5-6 could be allocated yielding no elongation. This demonstrates the inherent dependence of protection routing design on the particular layout of working path routes, a point noted by Coan et al who suggested joint optimisation of working and protection layouts to achieve a truly global optimal design (Coan et al, 1991) .

The remaining metrics, MMR and RCE, are shown in Figures 12 and 13, respectively. The estimate of database storage required per node shown in Figure 12, clearly indicates the deficit between failure dependent and failure independent rerouting strategies. Indeed, the deficit enlarges with the scale of the network, whereby source-based rerouting proves to be increasingly sensitive. Of course, it may be argued that a few kilobytes of memory is unimportant, however the estimates could be misleading. The reason for this is that a mean demand per node was computed, which is fairly artificial as some maximum value would be used in practice. Also, the storage would have to accommodate future physical and logical growth, since the assumption of a single VP between each node pair will often be unrealistic.

The computational effort needed to produce rerouting information is shown in Figure 13 with no distinction between the failure dependent schemes since only the number of required alternate routes was evaluated. If desired, suitable weighting of each scheme could allow individual curves to be fashioned, although this is not considered in this paper. The curve is striking as it highlights the sizeable computational overhead associated with failure dependent rerouting in contrast with its failure independent counterpart.

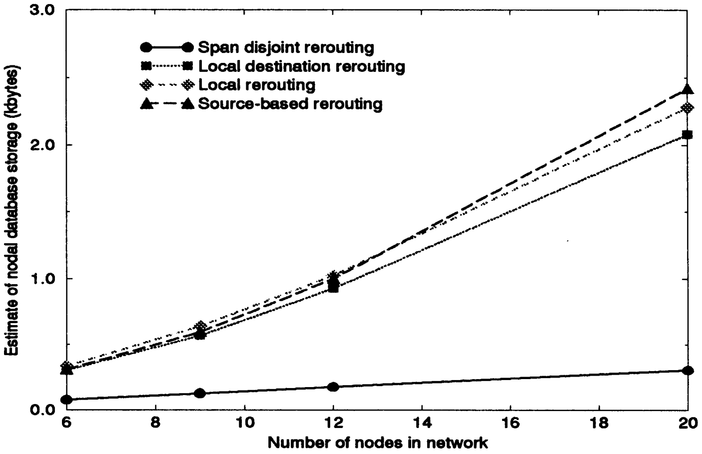


Figure 12: Mean Memory Requirements (MMR) versus network size

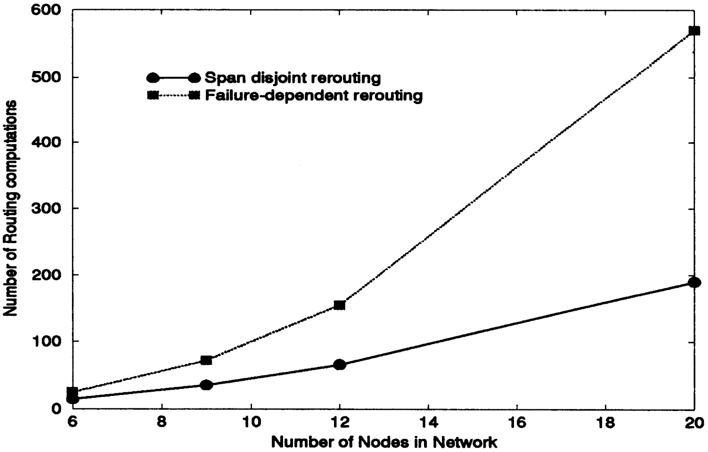


Figure 13: Routing Computational Effort (RCE) versus network size

4 DISCUSSION

4.1 Signalling protocol complexity

The preceding section presented a comparison of four pre-planned restoration schemes incorporating distinct rerouting policies, three of which are special cases of failure dependent rerouting, whilst the fourth constitutes the failure independent policy. An important facet of restoration which has not been discussed as yet, is the signalling protocol employed to activate pre-assigned routes. First, the speed with which restoration can be accomplished is paramount, since it governs the extent to which services will be adversely affected by the failure. In the AT&T paper (Anderson et al, 1994), no computer simulations of the signalling protocol to execute failure dependent restoration are described. Rather, an estimation of the restoration completion time for a 40 node network with modest processing time assumptions is cited as 58 msec. In (Veitch et al, 1995a), simulations of distributed protocols to realise failure independent backup path restoration, suggest that completion times of tens of milliseconds are possible. It can hence be postulated that comparable restoration completion times accrue with both methods of VP rerouting. Of additional concern is the ease with which protocols can be implemented. In the failure independent scheme, bidirectional F4 Operations, Administration and Maintenance (OAM) flows can be used to convey alarm and confirmation signals between the endpoints of the failed VP and the protection VP, respectively. This is a significant advantage given that certain OAM cells are already standardised (ITU-T, 1993b). With the failure dependent schemes, inter-nodal signalling channels, the properties of which have yet to be elucidated, must be employed to broadcast failure notification signals.

4.2 Planning adaptability and protocol robustness

The final issue to be considered as a basis for comparing the failure independent and failure dependent schemes, with these latter grouped as a whole, is that of robustness. First, we could analyse failure adaptability and question how each restoration scheme, in planning and execution, handles multiple span or node failures. With failure independent rerouting, transit node failures can be protected by allocating node disjoint protection paths with a suitable spare capacity allocation to match (Kawamura et al, 1994). No change to the signalling protocol is necessary with protection route activation performed in the same way as that for span failures, and no additional storage overheads are incurred. Unavailability of a protection route due to a multiple failure is easily identified with explicit confirmation of backup paths orchestrated from the endpoints (Veitch et al, 1995a). This could lead to a dynamic route searching protocol being invoked, or direct notification of the problem to a central controller. Planning for multiple span or node failures with failure dependent rerouting significantly impacts on the complexity of the whole approach. First, concerning the required planning effort, storage overheads and routing computation would increase sizeably due to the association of alternate routes with specific failures. This intractability would become accentuated with larger network topologies. Secondly, the signalling protocol would have to be modified so that nodes which receive broadcast messages glean an unambiguous picture of the current physical network topology.

The last matter of uncertainty which puts the alternative schemes to the test is the prospective lack of spare capacity in the network with which to support rerouted traffic. Although planning is performed in conjunction with spare capacity placement, or indeed in adherence with spare capacity constraints (Veitch et al, 1995c), occasions can arise where the supply will not meet the demand. If protection routes are activated under such circumstances, the quality of service of existing connections which share common buffer and transmission resources, and are unaffected by failure in the first place, could be unacceptably degraded. Because failure independent rerouting involves explicit confirmation of protection path capacity availability (Kawamura et al, 1994, Veitch et al, 1995a), if a path cannot be supported, the situation is quickly recognised and appropriate action taken. The problem with the failure dependent approach is that there is no notion of "capacity capturing" during crossconnect table activation, which is executed for a bundle of rerouted paths at any one time. This places a question mark over the supposed robustness of failure dependent rerouting.

5 SUMMARY AND CONCLUSIONS

This paper has highlighted the fundamental differences between two pre-planned VP restoration paradigms, the failure dependent and the failure independent methods. The choice of strategy influences implementation costs in terms of spare capacity, reserved VPIs, computational overheads and memory for rerouting information storage. Furthermore, the anticipated path elongation which impacts on the delay performance experienced by rerouted connections, must be accounted for. Metrics corresponding to all these factors were formulated, then, for a variety of grid network models, a comparative evaluation was carried out between the failure independent span disjoint rerouting scheme and three distinct failure dependent rerouting policies.

The span disjoint scheme required the least spare capacity for all networks considered, with the source-based rerouting version of failure dependent restoration a close second. The important point to note is that these results were not optimised, rather, a shortest hop routing algorithm was used throughout for comparative purposes. If optimisation was performed with the minimisation of a cost function based on spare capacity, an intuitive argument would suggest that the source-based failure dependent rerouting would require less spare capacity than the failure independent case. This is due to the tailoring of alternate routes to the actual failure, something which failure independent rerouting does not cater for. The other two failure dependent schemes, local-destination and local rerouting, displayed greater demand for spare capacity, with the latter being the "greediest", due to the frequent occurrence of *backhauling* meaning the same span is re-used in a route. Regarding VPI redundancy for protection routing, the outcome depends on whether or not VPI sharing is administered in the instances of failure dependent restoration. Without

VPI sharing, the failure independent scheme requires less idle VPIs than all the failure dependent methods, otherwise, it is the failure independent method that incurs the greatest redundancy. The degree of path elongation is minimised with source-based rerouting, whilst the span disjoint scheme improves over the other two failure dependent policies. The reason for the span disjoint scheme's inferiority to the source-based method in terms of path elongation, is that certain choices of working path routes forces the disjoint backup path to use a greater number of spans than is theoretically necessary. As expected, local rerouting was the most sensitive to path elongation effects, again due to backhauling. In terms of storage overheads and routing computational effort meanwhile, failure independent rerouting exhibits a clear advantage over all failure dependent schemes, with significantly less memory required and a computational effort which is proportional to the number of paths in the network only.

It is evident that in terms of required spare capacity, the number of spare VPIs, and the anticipated elongation of paths, the two most attractive solutions to pre-planned VP restoration appear to be the failure independent scheme and the source-based rerouting version of failure dependent protection. This latter should accomplish the lowest spare capacity provisioning if optimisation is performed, and furthermore, a smaller number of VPI numbers are idled. Also, for the network models considered, the path elongation was minimised with source-based rerouting. Regarding VPI redundancy, Kawamura postulated that the ratio of working to backup VPs in any link does not cause concern for VPI availability where disjoint backup paths are assigned (Kawamura et al, 1994). Although span disjoint rerouting demonstrated greater sensitivity to path elongation, this could be remedied by exercising a joint working/protection VP layout which minimises elongation effects. On the foundation of these observations therefore, it may be argued that the principal advantage of source-based rerouting is the prospect of spare capacity minimisation, though the computational effort needed to attain this, and how much gain over the failure independent scheme would accrue, remains open for investigation.

The potential advantage of source-based rerouting is offset by the distinct disadvantage of much greater storage overheads needed to support alternate routing plans. In addition, the routing computation will be far more intense for all failure dependent techniques compared with the failure independent approach. This combination of factors tends to swing in favour of the failure independent protection routing paradigm. This preference is consolidated by analysis of the qualitative issues related to protocol complexity and robustness. It was discussed in the penultimate section of the paper how backup path activation could be executed with simple OAM cell transmission protocols. These same protocols could be used whether a span or nodes fail. Indeed, to plan for node failures, node disjoint backup routes can be allocated, with no additional storage overheads incurred to support this mode of failure recovery. Furthermore, the confirmation of backup path availability allows detection of multiple failure or limited spare capacity conditions. All of these features of failure independent rerouting are in sharp contrast to the failure dependent approach which requires significant extra computation and storage space to accommodate other failures besides single span. The restoration protocol itself is complicated by unanticipated failures, and if there is limited spare capacity to support rerouted paths, there is no specified distributed mechanism to recognise the syndrome.

To conclude, the failure independent rerouting scheme for pre-planned Virtual Path restoration incorporates properties of resource efficiency, low implementation complexity and robustness, which combine to make it a suitable foundation for planning survivable ATM networks.

References

Anderson J., Doshi, B.T., Dravida, S. and Harshavardhana, P. (1994). Fast Restoration of ATM Networks. *IEEE Journal on Selected Areas in Communications*, January, 128-138.

Coan, B.A., Leland, W.E., Vecchi, M.P., Weinrib, A. and Wu, L.T. (1991). Using Distributed Topology Update and Preplanned Configurations to Achieve Trunk Network Survivability. *IEEE Transactions on Reliability*, October, 404-416.

ITU-T Recommendation I.311 (1993a). B-ISDN General Network Aspects.

ITU-T Recommendation I.610 (1993b). B-ISDN Operation and Maintenance Principles and Functions.

Kawamura, R., Sato, K-I. and Tokizawa, I. (1994). Self-Healing ATM Networks Based on Virtual Path Concept. *IEEE Journal on Selected Areas in Communications*, January, 120-127.

Sato, K-I., Ohta, S. and Tokizawa, I (1990). Broadband ATM Network Architecture Based on Virtual Paths. *IEEE Transactions on Communications*, August, 1212-1222.

Sosnosky, J. (1994). Service Applications for SONET DCS Distributed Restoration. *IEEE Journal on Selected Areas in Communications*, January, 59-68.

Veitch, P.A., Smith, D.G. and Hawker, I. (1995a). A Distributed Protocol for Fast and Robust Virtual Path Restoration. *Proc. 12th IEE UK Teletraffic Symposium*, Windsor, England.

Veitch, P.A., Smith, D.G. and Hawker, I. (1995b). Restoration Strategies for Future Networks. *Electronics & Communication Engineering Journal*, June, 97-104.

Veitch, P.A., Smith, D.G. and Hawker, I. (1995c). The Design of Survivable ATM Networks, in *Performance Modelling and Evaluation of ATM Networks, Volume 1* (Ed. D.D. Kouvatsos), 517-534. Chapman & Hall, London.

Wu, T-H. (1992). *Fiber Network Service Survivability*. Artech House.

Biographies

Paul Veitch is currently studying for a PhD degree in Electronic & Electrical Engineering at the University of Strathclyde, where, in 1993, he obtained his MEng degree with distinction in the same discipline. The research into ATM restoration techniques is co-sponsored by the EPSRC and BT Labs as part of a CASE award.

Geoffrey Smith is professor of communications at the University of Strathclyde. His research interests include management and control of broadband networks.

Dr Ian Hawker leads a team at BT research laboratories whose interests lie in modelling and performance evaluation of future networks.