Capacity Efficiency of Distributed Path Restoration Mechanisms in Optical Mesh Networks

Bart Rousseau and Fabrice Poppe

Alcatel, Network Architecture Dept., Network Strategy Group,
Francis Wellesplein 1, B-2018 Antwerpen, Belgium.
Tel.: +32-3-240 70 69, Fax: +32-3-240 48 88
{Bart.Rousseau, Fabrice.Poppe}@alcatel.be

Abstract. In this study the restoration performance of two closely related Sender/Chooser-based distributed path restoration protocols (both extensions of the Self-Healing Network (SHN) to path restoration) is compared. Some pathological situations in which these Sender/Chooser-based restoration algorithms perform suboptimally are identified and, where available, possible solutions are proposed. Built-in mechanisms to resolve contention between Sender/Chooser pairs were found to be very helpful. In addition, the performance of the distributed restoration algorithms studied was found to be close to its theoretical upper bound, suggesting that the pathological situations described may not be that important for real-life networks.

Keywords: distributed path restoration, Sender/Chooser based restoration protocols, optical mesh networks, survivable networks

1 Introduction

Although telecommunications networks have played a crucial role in society for some time now, the last decades society has become increasingly depended on the these networks. As a result the social and economical consequences of network outages have increased dramatically. With typically one cable cut per 0.003/km/year, cable cuts are surprisingly frequent. Given availability requirements of the order of 99.999% or higher (meaning that networks cannot be down for more than 6 min/year on average), it is clear that a fast and reliable recovery strategy is quintessential in contemporary networks.

Since most services can tolerate outages of up to 2 seconds [2], the goal is to develop restoration protocols that can achieve restoration in less than 2 seconds. The time required by a given restoration protocol to restore all failed paths (or as many as possible given the capacity constraints) is termed the restoration speed. It should of course be as low as possible and less than the 2 seconds goal put forward above. A good restoration protocol should combine a high restoration speed with a high restorability (the latter being defined as the percentage of

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 875–886, 2002.

failed paths that can be restored by the protocol under the existing network conditions). In addition the protocol should be capacity efficient and scalable.

Given the advantage of path restoration over link restoration in terms of capacity efficiency, it is surprising to find that relatively little work has been done on path restoration [3,4,5] as compared to link restoration [6,7,8,9,10,11,12,13, 14,15]. One promising approach for path restoration is the extension of the Self-Healing Network (SHN) protocol [12] to path restoration introduced by Iraschko and Grover [4]. The SHN and its extensions form a class of Sender/Chooser-based restoration protocols. In this study the restoration performance of two extensions of the SHN to path restoration is compared. As a result of this study some potential weaknesses of these protocols were identified. These will be illustrated in the first part of this work and, where available, possible solutions will be proposed. In the second part, our implementations of the protocol are used to verify what the consequences are of these issues in tightly dimensioned real-life networks. In the closing section conclusions are drawn.

2 Sender/Chooser Based Path Restoration Protocols

As already mentioned in the introduction, there exists a class of distributed restoration protocols that have as common characteristic the use of a Sender/Chooser mechanism. In this section a brief overview of the Sender/Chooser mechanism as applied to path restoration is given. Two variants of the basic protocol will be described. The first can be considered a direct extension of the SHN to path restoration. The second, introduced in ref. [4], is a modification in which an interference number is used as a measure to avoid contention for available resources. Only a brief description of these protocols will be given here, for more details the reader is referred to the original literature [2,4,12].

It is assumed that in case of a link failure, both the origin and destination nodes of all paths using the affected link are notified. When the terminating nodes of a path receive this notification, Sender/Chooser arbitration occurs, *i.e.* one node takes on the Sender role while the other takes on the Chooser role. This is done independently by both nodes using some arbitrary rule that yields a uniquely defined result (*e.g.* based on the ordinal number of the nodes). At this point the actual restoration process can start.

For each failed path for which a given node acts as a Sender, it sends a forward flooding (FF) message over each available outgoing link, thereby temporarily reserving spare capacity. These messages then propagate to the adjacent nodes. Such nodes that are neither Sender nor Chooser for a given path are called Tandem nodes (Note that a node can simultaneously be a Sender for one path, a Tandem node for another path and a Chooser node for yet another path. For any given path however any node is either a Sender, a Tandem or a Chooser node). The Tandem node rules discussed below, manage the contention amongst incoming messages for subsequent retransmission on the spare channels available at that node. Messages initiated from the node in its Sender role are integrated into the same overall competition. When a message reaches the Chooser node, this node answers with a reverse linking (RL) message. This messages traces back to the Sender node, meanwhile locking the required channels and releasing resources no longer required. The restoration path is locked and reserved once the RL message reaches the Sender node.

As mentioned above, for each failed path for which a given node acts as a Sender, it creates an internal arrival message for each available outgoing link. Each of these messages is assigned a locally unique index value. This permits tracebility in reverse linking to a specific port back at the Sender. These internal arrival message together with the external arriving messages (messages arriving over an incoming channel) compete for the available outgoing channels in order to get forwarded over one of these channels. In order to decide which messages are allowed to propagate and which are not, some kind of priority value is needed. One possibility is to use the repeat value (also called hop count) in order to discriminate between messages (this could be considered to be a direct extension of the Self-Healing Network to path restoration). In ref. [4] an interference number (IN) is introduced in order to avoid creating paths that could render a large number of other restoration paths infeasible. (A precise definition of the interference number will be given below). The authors propose to use this IN to decide which messages to forward.

Of each (Origin, Destination, Index) family (see ref. [4]), only one message, that with the lowest IN, is considered as a candidate for propagation. All the messages that are candidates (termed ‘precursors’ in ref. [4]) are then arranged in a table in order of increasing IN (breaking ties using the repeat count and then the index number if needed). Since the internal arrival messages, by construction, have an IN equal to zero, these will be at the top of the table. This table, in combination with the available capacity (number of outgoing channels) on each outgoing link, determines which messages will be propagated and which messages will be stopped in the current node. Next the table is examined from top to bottom, propagating each message over as many available outgoing links as possible using only one channel per message on any link (Note that although a given message may not be forwarded (or only partially) because the required capacity is not available, other messages further down the table might still be able to propagate). Messages are not forwarded over the link over which they arrived.

Some messages will not be able to propagate over all possible links. The number of messages that would have been forwarded on a link given enough capacity minus the number of messages that can be forwarded given the actually available capacity is called the interference number of that link (*i.e.* the number of messages ‘blocked’ by the messages that got through). When a message is forwarded over a link its IN is increased by the IN of that link.

3 Potential Shortcomings of Distributed Sender/Chooser-Based Path Restoration Algorithms

While working on our implementation of the protocol described in ref. [4] we identified a number of situations in which the Sender/Chooser-based restoration algorithms perform suboptimally, *i.e.* cases in which the final restoration ratios are lower than those achievable by centralized restoration algorithms. These will be discussed below and, where available, possible solutions are proposed.

3.1 Dependency on Topology

In ref. [4] reference is made to the so-called trap topology (see Fig. 1) to illustrate that a shortest path approach to restoration can in some cases lead to less optimal results than a maximum flow based approach. As an example suppose that two paths have been lost between nodes A and D. In this case, using a shortest path algorithm yields only one path (A-B-C-D) whereas using a maximum flow algorithm yields two paths (A-E-C-D and A-B-F-D). Neither using the repeat value nor using the IN in the protocol ensures that the two possible paths are obtained. This can be seen as follows. At a given moment in time, two messages will be received in Node C (one via A-E-C and one via A-B-C). Since these messages have identical IN and repeat values (an IN and repeat count of 1 and 2, respectively) the decision of which message to forward will only depend on the (arbitrarily assigned) index values of the messages. Depending on the relative magnitude of these index values either one or two paths will be obtained. In this case the restoration ratio is thus dependent on the arbitrary assignment of index values to the messages leaving the Sender node. Although this is a somewhat unsatisfactory situation, it is not immediately clear how this can be avoided. Note that this problem can occur if there is only one Sender/Chooser pair (such as *e.g.* in link restoration). Its occurrence simply depends on topological details and is unrelated to contention between restoration pairs. As such it is not specific to path restoration.

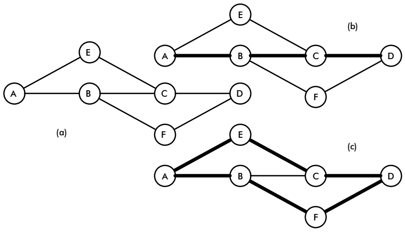

Fig. 1. Trap topology. Each link is assumed to have one spare channel available for restoration. Using a shortest path approach yields one path (*b*) whereas a maximum flow approach yields two paths (*c*). As explained in the text, which of these two possibilities is obtained using the protocols described above, depends on the arbitrary assignment of index values to the messages sent from the Sender node.

3.2 Unfairness towards FF Messages Close to the Chooser Node

The network illustrated in Fig. 2 is used in ref. [4] to illustrate the need for a protocol for path restoration that has some built-in means to avoid contention. Suppose that, in the network given in Fig. 2, each link in the network has a spare capacity of one channel. Further assume that we have lost 3 paths between nodes 1 and 8 and 4 paths between nodes 2 and 5. Nodes 1 and 2 are chosen to be the Sender nodes and nodes 5 and 8 the Chooser nodes.

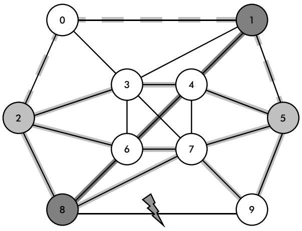

Fig. 2. Of the five possible paths only four are found due to the priority given to internal arrivals (see text for details).

The optimal solution in this case contains five paths (1-4-6-8, 2-0-1-5, 2-3-4-5, 2-6-7-5, and 2-8-7-9-5). However, using the protocol described above, only four paths (1-4-6-8, 2-3-4-5, 2-6-7-5, and 2-8-7-9-5) will be obtained. The reason that the path 2-0-1-5 is not found can be understood as follows. Node 1 is Sender node for four paths and as such will try to propagate four FF messages over each of its links (but due to the capacity constraints can send only one). When at a given moment in time a FF message sent by Node 2 arrives in Node 1 via node 0, it will be stopped there because it has a higher repeat value than the internal arrival messages of node 1 (that by definition have a repeat count of zero). Introducing an IN does not improve the situation since the internal arrival messages have, by construction, an IN that is not higher than that of any other message. In addition, as before, it will always have a lower repeat value. The fact that not all possible paths are obtained is thus closely coupled to the precedence that is given to the internal arrivals in the Tandem node rules. Note that this problem is specific for situations in which there are several Sender/Chooser pairs simultaneously performing restoration (regardless whether the repeat count or the IN is used).

3.3 Deadlocks

The problem mentioned in the previous subsection is a specific case of the more general phenomenon that under certain conditions a deadlock can occur. Such a case is illustrated in Fig. 3. Because messages have a higher probability to propagate near their own Sender node (due to their lower repeat count) than other messages, they risk blocking other paths. In some circumstances (especially when a very limited amount of spare capacity is available) this can cause the restoration procedure to stop although a (large) number of paths is still feasible given the available resources. Although usage of the IN can in some cases alleviate this problem somewhat, it cannot be excluded completely.

3.4 Race Conditions in the RL Procedure

In ref. [4] it is argued that it is preferable not to give reverse linking (RL) absolute priority, *i.e.* to allow the RL procedure to legitimately fail en-route. However, if no special measures are taken (requiring, as will be shown below, an extension

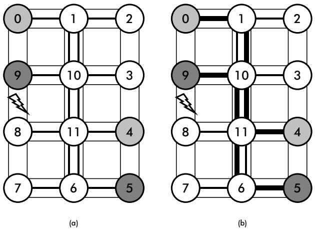

Fig. 3. Illustration of a deadlock situation in which none of the possible paths are found. Messages from Sender node 5 are stopped in node 10 because they have a larger repeat value than those emitted by node 0. For the same reason, messages from Sender node 0 are stopped in node 11.

of the protocol) this can lead to erroneous results in which a restoration path is believed to exists, where in fact only a part of this path is available. This can be illustrated using the following example (see Fig. 4).

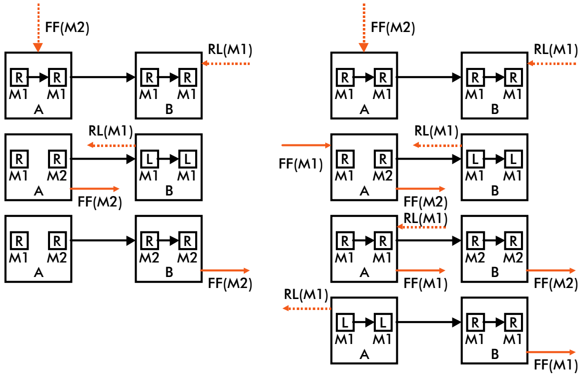

Fig. 4. Successful (*left*) and erroneous (*right*) cancellation of a RL procedure (see text).

The initial situation is such that a message M1 has traced a path from its Sender node to its Chooser node through Tandem nodes A and B. Now suppose that before the RL message corresponding to M1 arrives back at node A, the following occurs. The port over which M1 was sent from node A to node B is taken over by another message (M2). Then, before M2 arrives at node B, M1 reclaims the port (because *e.g.* the precursor of M2 disappears). This means that when the RL message corresponding to M1 arrives back at node A, it will be propagated in the direction of the Sender node, because a mach between a FF and an RL message was found. When this RL message arrives at the Sender node, a complete locked-in restoration path is supposed to exist between the Sender and Chooser nodes. In reality however, only a partial path exists since the message M2 will have torn down the part of the path closest to the Chooser node (*i.e.* between node B and the Chooser node).

To solve the problem mentioned above, we need to be able to ensure that the port has not been taken over at any time before the RL message arrives. For this purpose a locally unique identifier should be assigned to each message when passing through a port. This enables the RL message later on to verify that the port was not taken over by another message between the time the that the original FF message left this port and the time the RL message arrives in this port. In our current implementation a time stamp is used for this purpose, but any locally unique identifier (per port) will do.

3.5 Hold-off Time

Suppose that in the network given in Fig. 5 two paths have been lost, one between nodes 5 and 14, and one between nodes 4 and 6. Given a spare capacity distribution as shown in Fig. 5 both paths can in principle be restored. Using only the repeat value in the protocol described above, only one path will be obtained (5-9-10-14). One may be tempted to think that using the IN will result in two paths. That this is not the case can be understood as follows. Lets call the path from node 5 to node 14 P1 and that from node 4 to node 6 P2. The messages involved in the restoration of these paths can then be identified as M1 and M2 for paths P1 and P2, respectively. The first event that occurs is the arrival of M1 in node 9. Since it is the only message present at that time in this node for forwarding on the link from node 9 to node 10 and one spare channel is available on this link, the message will be forwarded over this link with an IN equal to zero. Shortly after, M1 arrives in node 9. We now have two messages to forward over the link 9-10 and only one channel available for restoration. This means that only one message will be forwarded and that it will have its IN incremented by 1. Since both messages arrive with an IN of zero, but M1 has a lower repeat count (1 for M1 versus 2 for M2) M1 will be propagated over link 9-10 with an IN of 1. As a consequence, node 1 first receives M1 (via 5-9-10-14) with an IN of zero. Node 14 reacts by sending an RL message in the direction of node 5 (via the link 14-10). However, before the RL attempt can succeed, the path is taken over by M1 with an IN of 2. When this message arrives in node 14 (again via 5-9-10-14), a new RL attempt will be started. Note, that

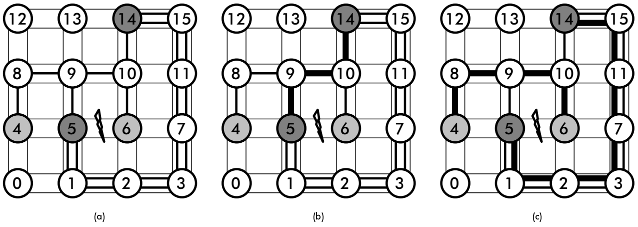

Fig. 5. Using just the repeat value, only a single path is found (*a*). Using the IN, either one (*b*) or two (*c*) paths are found, depending on whether or not a hold-off time is used.

at the same time, a message M1' is tracing an alternative restoration path for P1 (5-1-2-3-7-11-15-14). However, since node 14, by the time M1' arrives, has already responded with an RL event to M1, it will ignore the arrival of M1' even though this message has a lower IN (1 for M1 and 0 for M1'). Therefore, when the RL event, started upon the arrival of M1 in node 14, succeeds, only one path is found (5-9-10-14).

This can be solved by introducing a hold-off time before which no RL event is allowed to start. Two scenarios are possible. Either the hold-off time is applied after the reception of the alarm signal or it is applied after the arrival of the first FF message for the given path (Note that this is somewhat similar to the use of a hold-off time in the FITNESS algorithm [15,16]). The result of introducing a hold-off time is that (at least when the hold-off time is chosen sufficiently long) node 14 will not start any RL attempt until both M1 and M1' have arrived. Since M1' has a lower IN than M1, node 14 will use M1' for RL. When this RL message sent from node 14 arrives back in node 5, a restoration path for P1 is found (5-1-2-3-7-11-15-14). In addition, upon the arrival of the RL message in node 5, all remaining messages for P1 will be canceled. As a consequence link 9-10 will become available for restoration of P2. Thus, introducing a hold-off time ensures that the IN can fully play its role. As a net result in this case, two paths instead of only one are obtained.

3.6 Dynamicity Due to IN

One of the major consequences of introducing the interference number is the increased ‘dynamicity’ of the protocol. The repeat count of a message that has arrived in a given node through a given path does not change in time. The IN however can change due to events taking place in any of the nodes the message passed through on its way to its current location (in fact the IN of the message does not change but more recent messages for the same path but with a different IN will take over the ports in use by the original message). This increased dynamic behavior can in some circumstances (especially in larger networks with a small amount of spare capacity) dramatically slow down the restoration process (in some cases even to the extent that no restoration paths are found within the two seconds time limit). We have therefor decided in our implementation of the protocol to reduce this ‘dynamicity’ as follows. Suppose that a message has traced a path from its Sender node in the direction of its Chooser node via Tandem node A. In our implementation, when the broadcast pattern in node A changes (because *e.g.* a new message has arrived) the old message is not replaced by a new message with an updated IN. This is a different approach than that used in the original description of the protocol [4]. Letting messages that have already been sent keep their INs, even when they should have been updated due to changes in the local forwarding pattern, reduces the dynamic behavior described above. However, at the same time, this limits the extent to which the IN can fulfill its role of avoiding the creation of unfavorable paths.

4 Simulation Results

In this part the results obtained using our implementation of the protocol described above will be discussed. The performance of this algorithm in tightly provisioned networks is studied. The results indicate that the phenomena described in the previous part are not that important for real-life networks.

4.1 Parameters

Messages with a size of 512 bits are sent over a dedicated signaling channel having a bit rate of 152 Mbps. The propagation speed on links was set to 2 10^5 km/s. A processing delay of 0.5 ms was used incremented with 1ms when the tandem logic required reevaluation of the composite forwarding pattern.

4.2 Networks

In the simulations three networks (with 13, 19, and 30 nodes, respectively) were used in combination with two demand patterns for each network (see Table 1 for details). These networks, of which one is depicted in Fig. 6, have a relatively high nodal degree and are therefore representative of core networks. The networks only have the minimum amount of spare capacity required to allow for 100% restorability for all single link failures in a revertive restoration scenario, *i.e.* without release of the capacity used by the affected lightpaths prior to restoration (stub release). The method that was used to place spare capacity was described in ref. [17]. It does not take into account the details of the distributed restoration algorithm. As such, the networks studied can be used as benchmarks: because of the spare capacity placement, optimized centralized restoration mechanisms can achieve a restoration ratio of 100% for every single link failure. Through simulations we studied how close to this upper bound the performance of the distributed restoration mechanisms is.

Table 1. Number of nodes (N), number of links (L), average nodal degree (n), average span length in km (l), number of origin/destination pairs ($N_{O/D}$), total demand for all origin/destination pairs (D), average number of working channels per link (W), average number of spare channels per link (S) and network redundancy (R) for the different networks used in this study.

	N	L	n	l	$N_{O/D}$	D	W	S	R
I13S	13	24	3.7	105	31	57	4.2	3.5	0.85
I13L	13	24	3.7	105	67	209	16.7	10.3	0.62
E19S	19	40	4.2	779	78	132	6.3	2.9	0.46
E19L	19	40	4.2	779	110	246	11.7	4.8	0.41
I30S	30	59	3.9	146	114	261	10.5	6.4	0.61
I30L	30	59	3.9	146	301	972	45.7	23.5	0.51

4.3 Results and Discussion

The restoration ratio and restoration time obtained for all possible single-link failures in the networks described above are summarized in Table 2, both with (non-revertive) and without (revertive) stub release. Note that, despite the tightly optimized amount of spare capacity, high restoration ratios are obtained, both using the repeat count and using the IN. For the non-revertive approach, in all cases where less than 100% restorability was achieved, this was due to the path sets chosen by the algorithm and not due to *e.g.* deadlocks such as described above. In the non-revertive case using the interference number has a negligible effect. In the revertive approach, however, use of the interference number clearly yields better restoration ratios, *e.g.* for the 30-node network, 86% of the 5390 lost paths could be restored using the repeat count, whereas using the INs this increased to 94%. Note that although for all the networks the spare capacity was dimensioned to allow for complete recovery from all single-link failures using a revertive approach, a restorability of 100% was not obtained in all cases using the distributed restoration protocol.

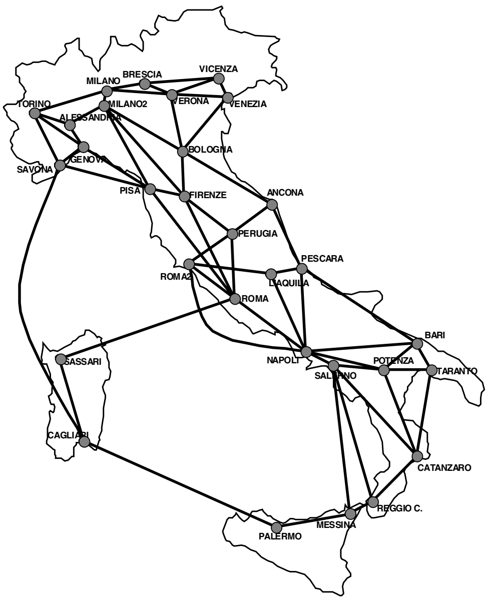

Fig. 6. One of the networks used in this study. The network characteristics are summarized in Table 1.

In addition to the single link failure simulations discussed above some double link failures where performed using the 13-node network. Also in this case does using the INs yield improved results compared to using only the repeat count. Of the 2820 failed paths only 2014 could be restored using the repeat count whereas

Table 2. Number of restored paths ($N_{restored}$), number of failed paths (N_{failed}), time at which the last restoration paths was found (t_{max}, in ms), average restoration time (t_{avg}, in ms) and amount of spare capacity used for restoration paths (SC_{used}).

		No stub release	*Stub release*			
	N_{failed}	$N_{restored}$	$N_{restored}$	t_{max}	t_{avg}	SC_{used}
IN						
I13S	200	200	200	39	30	16
I13L	800	800	800	61	39	24
E19S	500	498	500	133	67	21
E19L	938	935	938	162	73	22
I30S	1236	1223	1230	77	44	16
I30L	5390	5078	5388	126	56	18
Repeat count						
I13S	200	200	200	37	30	15
I13L	800	800	800	57	38	23
E19S	500	498	498	118	60	19
E19L	938	931	938	144	64	21
I30S	1236	1208	1235	67	39	15
I30L	5390	4623	5390	115	53	18

2080 could be found using the interference numbers (note that the network was only designed to handle single link failures).

5 Conclusions

In this study the restoration performance of two closely related distributed path restoration protocols, both extensions of the SHN, is compared. The first can be considered to be a direct extension of the SHN to path restoration. The second, based on an interference heuristic, has a built-in measure to avoid contention for spare resources. In the fist part a number of situations were identified in which Sender/Chooser-based restoration protocols clearly will perform suboptimally and, where available, solutions were proposed. In the second part the restoration behavior of the two restoration protocols is examined using simulation on tightly dimensioned real-life networks. Built-in mechanisms to resolve contention between Sender/Chooser pairs were found to be very helpful. In addition it was found that the restoration performance of the distributed restoration algorithms studied was close to its theoretical upper bound, suggesting that the pathological situations we discussed may not be that important for real-life networks.

References

1. Doucette, J., Grover, W.D.: Influence of Modularity and Economy-of-Scale Effects on Design of Mesh-Restorable DWDM Networks. IEEE Journal on Selected Areas in Communications **18** (2000) 1912–1923

2. Grover, W.D.: Distributed Restoration of the Transport Network. In: Aidarous, S., Plevyak, T. (eds.): Telecommunications Network Management into the 21st Century. IEEE Press (1994) Chapter 11
3. Doshi, B. T., Dravida, S., Harshavardhana, P., Hauser, O., Wang, Y.: Optical Network Design and Restoration. Bell Labs Technical Journal January-March (1999) 58–84
4. Iraschko, R.R., Grover, W.D.: A Highly Efficient Path-Restoration Protocol for Management of Optical Network Transport Integrity. IEEE Journal on Selected Areas in Communications 18 (2000) 779–794
5. Wuttisittikulkij, L., O'Mahony, M.J.: Use of Spare Wavelengths for Traffic Restoration in a Multiwavelength Transport Network. Fiber and Integrated Optics **16** (1997) 343–354
6. Chow, C.E., Bicknell, J.D., Mccaughey, S.: Performance analysis of fast distributed link restoration algorithms. Int. J. Commun. Syst. **8** (1995) 325–345
7. Chujo, T., Komine, H., Miyazaki, K., Ogura, T., Soejima, T.: Distributed self-healing network and its optimum spare capacity assignment algorithm. Electron. Commun. Japan. Part 1. **74** (1991) 1–8
8. Doverspike, R., Sahin, G., Strand, J., Tkach, R.: Fast Restoration in a Mesh Network of Optical Cross-connects. Proceedings of the Optical Fiber Communications Conference, OFC '99 (1999)
9. Ellinas, G., Hailemariam, A.G., Stern, T.E.: Protection Cycles in Mesh WDM Networks. IEEE Journal on Selected Areas in Comm. **18** (2000) 1924–1937
10. Lee, H., Song H.-G., Chung, J.-B., Chung, S.-J.: Preplanned rerouting optimization and dynammic path rerouting for ATM VP restoration. Telecomm. Syst. **14** (2000) 243–267
11. Gersht, A., Kheradpir A., Shulman A.: Dynamic Bandwidth-Allocation and Path-Restoration in SONET Self-Healing Networks. IEEE Transactions on Reliability **45** (1996) 321–331
12. Grover, W.D.: Selfhealing networks — A distributed algorithm for k-shortest link-disjoint paths in a multi-graph with application in real-time network restoration. Ph.D.-thesis. University of Alberta (1989)
13. Poppe, F., De Neve, H., Petit, G.H.: Constrained Shortest Path First Algorithm for Lambda-Switched Mesh Optical Networks with Logical Overlay Och/SP Rings. 2001 IEEE Workshop on High Performance Switching and Routing (2001) 150–154
14. Sakauchi, H., Nishimura, Y., Hasegawa, S.: A self-healing network with an economical spare-channel assignment. Proc. IEEE Globecom'90 (1990) 438–443
15. Yang, C. H., Hasegawa, S.: FITNESS: Failure Immunization Technology for Network Service Survivability. Proc. IEEE Global Telecomm. Conf., Globecom '88 (1988) 1549–1554
16. Bicknell, J., Chow, C.E., Seyd, S.: Performance Analysis of Fast Distributed Network Restoration Algorithms. Proceedings IEEE GLOBECOM '93 (1993) 1596–1600
17. Poppe, F., Demeester, P.: Economic Allocation of Spare Capacity in Mesh-Restorable Networks. Proc. Of the Sixth International Conference on Telecommunication Systems, Modeling and Analysis (1998) 77–86