46

Integration of Distributed Restoration Procedures in the Control Architecture of ATM Cross-Connects

Hans Vanderstraeten, Dominique Chantrain, Gzim Ocakoglu, Leo Nederlof, Luc Van Hauwermeiren
Alcatel Corporate Research Centre, Location Antwerp
Francis Wellesplein 1, B-2018 Antwerpen, Belgium, tel. +32/3-240.79.05, fax +32/3-240.99.32 E-mail : hvds@rc.bel.alcatel.be

Abstract

It has been claimed that Distributed Restoration Algorithms (DRAs) have considerable advantages in terms of speed compared to centralised restoration schemes. However, these statements have always been based on simulations of the algorithms only. In order to study DRAs in a real environment, a testbed for experimenting with restoration in ATM networks has been realised. The testbed consists of five 8x8 STM1/ATM cross-connects, and is supervised by a central network management system. Hereby, provisions have been made to incorporate a maximum of flexibility, in order to test various restoration mechanisms, while striving for optimal performance.

The DRA has been integrated into the testbed environment and its robustness has been demonstrated for various network configurations of the testbed. Early results show that it is indeed possible to find the alternative routes within the call dropping time limit (± 2 seconds).

Keywords

Network survivability, distributed restoration algorithms, ATM cross-connects

1 INTRODUCTION

In broadband networks, more and more traffic is concentrated on fewer network elements and fibre routes. This implies that network outages (cable cuts, switching system failures, ...) will cause huge losses of revenue for operators, service providers and commercial users. Therefore, network survivability is essential for broadband systems.

Various approaches for survivability exist (Wu, 1995; Nederlof, 1995). Automatic protection switching is the most simple approach, and is commercially available today. This approach is

L. Mason et al. (eds.), *Broadband Communications*

also used in line switched ring systems. Path protection switched SONET/SDH rings switch to the backup ring on the SONET/SDH path level.
Recently, distributed restoration techniques received considerable attention. Although potentially slower than protection switching to preplanned dedicated spare resources, they have the advantage to operate on meshed and therefore also flexible topologies. They start from the assumption that each node has only limited knowledge of the network topology. Storing too much knowledge at each node might cause problems of consistency among distributed databases when changes occur in the network. Therefore, the distributed restoration algorithms build up network information (topology and location of spare resources) required to restore the failed paths *after* the failure has occurred, guaranteeing an up to date view of the network. Although the gathering of information before restoration consumes time, all proposed distributed restoration algorithms have the explicit ambition to prevent call dropping.
Grover et al. presented the first distributed restoration algorithms (Grover, 1987; Grover, 1990), later followed by several other research groups (Han Yang, 1988; Komine, 1990; Chow, 1993a; Chow, 1993b). All these algorithms are targeted at networks based on the SONET/SDH transmission standards. Kawamura et al. (1992) developed an algorithm for ATM VP networks, which they proved were more efficient than their STM counterparts due to the properties of ATM.
This paper reports on the development of a Distributed Restoration Algorithm (DRA) for ATM networks, and the integration of the algorithm in a network of real ATM cross-connects. Section 2 summarises the objectives of the research on survivability. As will be shown, these objectives will influence some decisions taken for the integration. Section 3 introduces briefly the DRA to be integrated in the ATM cross-connects. Its performance characteristics as derived by the simulation are shortly discussed. The ATM cross-connect and the testbed are presented in Section 4, while Section 5 discusses the integration of the DRA in the cross-connect architecture. Network management issues are shortly addressed in Section 6. The conclusion summarises the current status of the research, including early results on the testbed, and gives an overview of the issues still to be investigated.

2 OBJECTIVES OF RESEARCH ON NETWORK SURVIVABILITY

Up to now, the performance of distributed restoration algorithms has only been analysed using simulation tools (Bicknell, 1993). The principal parameters in which performance is expressed are the spare capacity efficiency and the speed of restoration. All published studies focused on the performance characterisation on network level. Nodes were considered as black boxes, and the contribution of the internal delays caused by the message processing and cross-connecting were considered as being well defined and fixed. However, it is clear that, given the inherent complexity of STM and ATM cross-connects, both from a hardware and software point of view, this reasoning has to be refined if reliable statements are to be made on the restoration speed of any algorithm in a real network. Recently, Wu and Kobrinski (1993) first expressed concerns on the network simulation-only approach. These authors studied internal STM cross-connect architectures, and showed that optimising overall restoration speed on network level requires an optimum performance of all relevant functionalities inside the cross-connect. This conclusion was confirmed by a study of Kobrinski and Azuma (1993).

An additional shortcoming in current research is that the interaction of any DRA with network management has not been studied in depth. Yet it is of paramount importance that DRAs, although operating autonomously, i.e. independently from the network manager, can be *controlled* by the network management. For instance, the network management has to be able to validate the DRA before it is authorised to take decisions autonomously, and to select a specific DRA out of a set of different protection and restoration techniques/mechanisms, depending on the desired trade-off between the advantages/disadvantages of any DRA for the specific network configuration that is managed.

This paper reports on the integration of a DRA in an ATM cross-connect system. This work is part of a research project on network survivability which has as major objectives :

1. to realise an ATM network platform for experimenting with restoration,
2. to prove that DRAs actually do work in *real* ATM networks, consisting of *real* ATM cross-connects,
3. to allow realistic simulations on existing and future networks of operators, through validation of the node and network models,
4. to determine the characteristics of different proposed restoration schemes (backup VP, span-based DRA, centralised restoration, ...) in real ATM networks,
5. to develop and validate the integration of the autonomous DRA with centralised network management.

These objectives have implications on the specific choices made for the integration. The first objective implies that the integration should for instance allow flexible inclusion of different algorithms in the node, that a general, flexible messaging scheme is preferred over an optimised one (single cell ATM OAM messages), and that the means have to be included to measure the behaviour of the restoration at any time (through inserting/extracting timestamps).

The second and third objectives are realised by building the testbed on *commercially available* ATM cross-connects. The specification of the DRA is first validated using the simulation environment. In a next stage, the DRA is integrated into the ATM testbed. Using the network manager, different test scenarios are set up and executed on the network. Additionally, the node model used in the simulation is brought in line with the ATM cross-connect of the testbed (the DRA has to show the same behaviour in terms of several different variables like speed of restoration, sequence of alternative routes found, number of messages generated, ..., on both the testbed and the testbed representation in the simulation tool). In this way, extrapolations to large ATM networks become reliable.

The fourth objective requires that the integration should allow each of the restoration procedures to perform optimally. The first objective will preclude optimal performance, so that measures have to be taken to allow relevant and relatively comparable benchmarking.

The last objective requires that a communication mechanism is foreseen between the node, including the DRA, and the Network Management.

3 DISTRIBUTED RESTORATION ALGORITHM

The Distributed Restoration Algorithm (DRA) is triggered by alarms due to failures of links, and searches for new routes around the failure in a joined effort with the other nodes in the network. Here, we will only explain the general principles of the DRA. For more details, we refer to Nederlof (1995). Extensive simulation results can be found in Vanderstraeten (1996). The proposed algorithm is designed to cope with single link, multiple link, and node failures.

3.1 Working principle

The route search process is based on a flooding of *Request messages*, originating from the *Request Source* (RS) nodes adjacent to the failure as in Figure 1. Intermediate nodes, referred to as *Tandem* (T) nodes, store the request, and forward updated copies of the Request message. As these messages propagate between the nodes, spare capacity that is requested will be explicitly reserved for each message until this capacity is captured as part of an actual alternative route, or released to make place for a different alternative route. A limit is set to the flooding procedure by specifying the maximum number of tandem nodes by which a request message can be remote from the original RS node. This parameter is further referred to as the *hop count*.

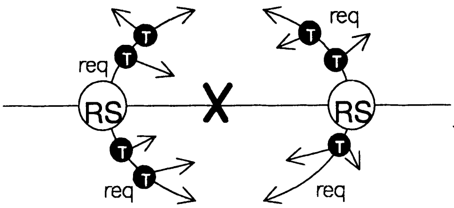

Figure 1 Request message flooding.

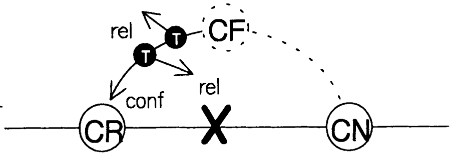

Figure 2 Confirmation.

When a route is found, i.e. when request messages meet somewhere between the two RS nodes, the Tandem node where the branches meet, now labelled *Confirm* (CF) node, designates the RS node with lowest ID as *Chooser* (CR) and the other as *Chosen* (CN). The Confirm node sends *Confirm messages* down the candidate alternative route towards the Chooser as illustrated by Figure 2. Any surplus bandwidth reservations are cancelled implicitly, or by sending *Release messages* on obsolete request branches.

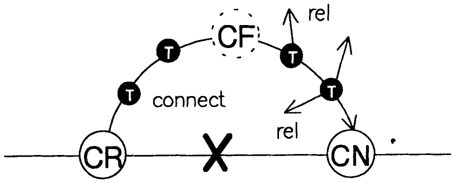

Figure 3 Reconfiguration.

Figure 3 shows the Chooser node that makes a selection of the VPs that can be allocated to candidate alternative routes. Information about the re-routed connections is sent along the route to the Tandem nodes and the Chosen node in *Connect messages*. Each Tandem node between the CF node and the CN node now releases obsolete request branches.

3.2 Simulation results

The proposed DRA has been developed and validated on a discrete event simulator, in which a network model and a node model have been integrated.

The network model for which the results are reported here, is shown in Figure 4. The model has been used by several other authors (Bicknell, 1993). The figures on the links indicate the capacity working(spare). The connections are assumed to have unit bandwidth. In our simulations we assume that each connection corresponds to a single VP.

By means of a node-level simulation, a suitable node model has been derived in order to describe the aggregate delay caused by the processing and queuing of restoration messages inside the nodes. In the presented simulation results, the message delay on the input and output of each node in the network is taken to be statistically distributed with a mean value between 10 and 20 msec each.

Table 1 illustrates as an example the restoration of the link failure between nodes 4 and 8 of Figure 4. The link originally carried 59 (bi-directional) active connections, which are restored by using the spare capacity in the network and divided among 5 alternative routes. The table also provides the restoration time for each alternative route found in the network (i.e. the time between the detection of the link failure and the arrival of the connect message in the Chosen node). The restoration process is completed for this link failure after 430 msec, which means that no more restoration messages are passing in the network and every extra bandwidth reserved during the restoration process, which does not result in an alternative path, is released after this time. For this particular link failure, the original route between the nodes 4 and 8, is replaced by a set of alternative routes, which carry connections via multiple hops instead of a direct link between two endpoints. The restoration results in an average of 4.71 hops in the connections.

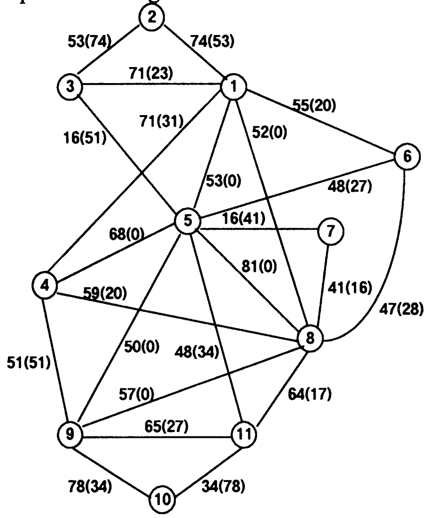

Figure 4 The reference network model used for simulating the distributed restoration algorithm.

Figure 5 shows the restoration times (complete restoration) for all single link failures in the reference network. The complete duration of the restoration process varies from 180 msec (link failure on 5-7) to 490 msec (link failure on 5-11). The time until the last alternative route is connected varies from 123 msec (link failure on 1-3) to 445 msec (link failure on 5-11). The number of hops has been set in all cases large enough to allow 100 % restoration. The average number of hops on the alternative routes in this analysis is 3.61, while the average restoration time was 303 msec.

It should be emphasised that in these simulations the node model is not comparable yet with the node representation of the real ATM cross-connect, but is more compliant to the model used in other simulations in literature, allowing a meaningful comparison between algorithms (Vanderstraeten, 1996).

Table 1 Link failure restoration between nodes 4 and 8 of Figure 4.

alternative route (indicated by node id)	number of VP connections	restoration time (msec)
4-9-11-8	17	118
4-1-6-8	20	141
4-9-11-5-7-8	10	222
4-9-10-11-5-7-8	6	276
4-9-10-11-5-6-8	6	386
	total = 59	completed after 430 msec

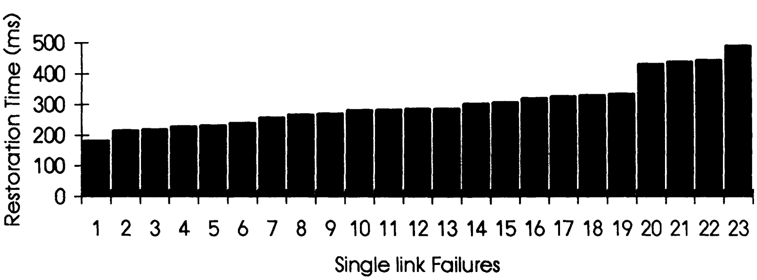

Figure 5 Restoration times of all single link failures on the network of Figure 4.

4 THE TESTBED NETWORK

The ATM cross-connect node used to configure the experimental testbed is an adaptation of the original commercially available cross-connect. A key aspect of this cross-connect is its modularity. This aspect has been fully applied to configure the experimental testbed. Therefore, we first explain the switching and control architecture of the original cross-connect. Afterwards the adaptations and their implications for our testbed application are described.

4.1 The ATM cross-connect

The ATM cross-connect used in the testbed is based on a multipath self-routing switching principle and an internal transfer mode using multislot cells. This switching concept is shown to have a number of advantages (Boettle, 1990), among which the modular extendibility from small to very large systems.

The Termination Link Board (TLK) is the part of the switch that is dependent on the external environment. It performs termination of the external transmission system, the processing required for the external transfer mode (ATM in our case), the conversion between ATM cells and multislot cells, the traffic distribution to and routing from the switching fabric, and resequencing (Banniza, 1991). The board terminates eight STM1/ATM links, and contains, next to the termination logic, also an On-Board Controller (OBC). An on-board mechanism is foreseen to insert/extract cells to/from an external ATM cell stream to/from the OBC. Distribution to and routing from the switching fabric is done using two integrated switching elements. As such, the TLK contains the first stage of the switching fabric.

Subsequent stages of the switching fabric are realised using Switching Modules (SMs), consisting of eight integrated switching elements, structured in a two-stage arrangement. Again, each SM is equipped with an OBC.

Control communications are needed between individual OBCs and the control station (CS) handling call control for services as well as OAM functions at system level. For proper structuring of control communications and loose coupling between transport network and control applications, an intermediate level of control resource has been introduced, the Rack Configuration Controller (RCC), distributed per rack and dedicated to all front-end control functions relating to the OBCs equipped in that rack, but also the power supply and cooling functions.

4.2 The control architecture

Figure 6 shows the control architecture of the ATM cross-connect.
The control functions running on the OBCs of the SMs and TLKs include the initialisation, maintenance and traffic load management. Furthermore, the TLK OBCs fill in the VPI/VCI translation tables, and perform connection control functions (among which is policing).

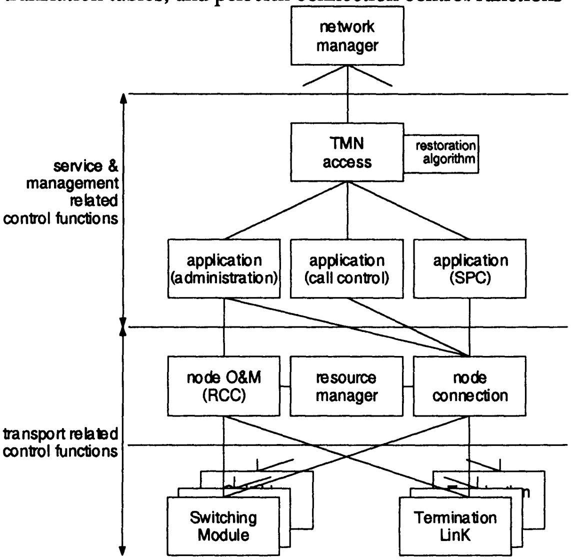

Figure 6 A generic representation of the control architecture of an ATM cross-connect.

The *node O&M* module runs on the RCC, and provides all functionality to allow the RCC to autonomously initialise and monitor the actual hardware equipment of its rack. Maintenance functions in the RCC manage the operational configuration available for traffic. The latter depends on various alarm conditions at rack level and on operational status conditions reported by each OBC about its own board functions and input/output links.
The *node connection* module runs on the CS, and performs resource allocation (VPI/VCI values, and bandwidth) and connection control functions. This module gets its information on free resources from the *resource manager* module (running on CS), which maintains an up to date view on all operationally available resources through the node O&M module.
The *application* modules implement common services offered by the switch. The *Semi-Permanent Connection (SPC)* module appends a time scheduler to a cross-connect point, while the *Administration* module defines external access ports, and maintains user profiles.
The *TMN access* module implements a CMISE interface, and translates the messages and relays them towards the appropriate service modules.
In this way, the control architecture of the cross-connect can be organised as a chain of interacting blocks representing the flow of O&M-related information from the lowest level up to the highest level (being the access for TMN), and a chain of blocks passing connection-related information up to the highest level.

4.3 The ATM cross-connect node in the testbed

The ATM cross-connect node used in the testbed is an adaptation of the original commercially available cross-connect, the latter being described in detail above. The only adaptation consists of reducing the hardware of the cross-connect to the bare minimum, while retaining the complete control architecture, thus exploiting the modularity property of the MPSR switching concept. Since the TLK includes *both* termination and switching functionalities, this board in principle suffices to switch ATM. However, since the control architecture should remain unchanged, the software module running on the RCC is moved to CS. In this way, all maintenance functions to be performed by the RCC still are performed, be it by the CS.
By only reducing the hardware, and preserving the original control architecture, the internal *and* external behaviour of the cross-connect are identical to the commercial version, except for the fact that the size of the switching fabric is reduced to the minimum, and the hardware support functions that are not strictly needed for the switching functionality (such as the control of periphery like fan units and the like) are removed. In this way, the objectives outlined in Section 2 are satisfied, i.e., the testbed allows to study the behaviour of the DRA in terms of speed of restoration and spare capacity usage in circumstances that are directly comparable to the commercially run ATM networks.

4.4 The testbed

The testbed set up for experimenting with restoration consists of five single TLK ATM cross-connects (Figure 7), each terminating eight input/output STM1/ATM links. This arrangement allows the topology to be fully meshed.

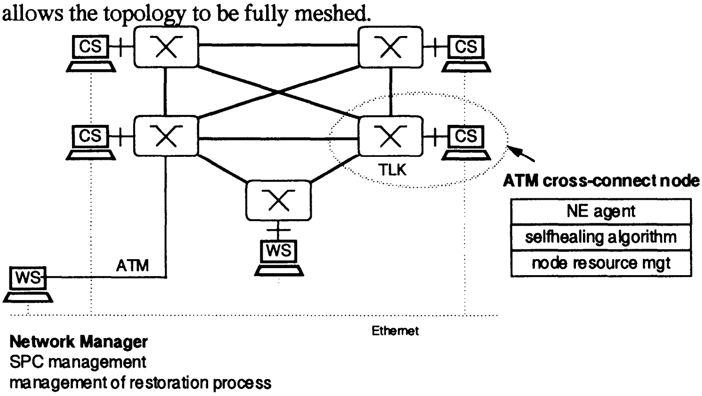

Figure 7 The testbed for experimenting with restoration, consisting of five 8x8 ATM switches, managed by a central network management application.

Communications between the TLK and CS is based on a (point-to-point) proprietary internal control protocol, while communication between the network manager and the agents (physically the CSs) can be either through Ethernet, or through ATM.

Of course, the integration of the DRA in the control architecture, and its interaction with the centralised network management, will require further adaptations. These are extensively discussed in the subsequent sections.

5 INTEGRATION OF THE DRA IN THE CONTROL ARCHITECTURE OF AN ATM CROSS-CONNECT

The choice between possible alternatives for the integration will be determined by what information the DRA needs from its external environment, and by the objectives pointed out in section 2, i.e., as close as possible to real configurations, but allowing experiments.

5.1 DRA's requirements from its external environment

As far as the integration is concerned, only the external information that the DRA needs for proper operation is important, and the mechanism to transfer messages to the neighbouring nodes and the network management have to be considered.
The DRA needs the following information from the node (Nederlof, 1995) :

- the number of external links,
- the nominal bandwidth of each of the links,
- the connections (VP/VC/bandwidth) on each of the links,
- the used and spare bandwidth on each of the links.

From the network manager, the DRA has to obtain the following items :

- its network wide node-id,
- the node-ids of the nearest neighbour nodes,
- a network wide unique id for each of the connections that go through the node,
- the node-ids of the one but nearest neighbour node in both directions for each connection individually.

For the communication between the DRAs a messaging system has to be integrated in the testbed. Obviously, the communication should be optimal in terms of speed. Ideally, the DRA messages should be short enough to fit in a single cell. The intrinsic OAM cell flow foreseen in the ATM standard is the first candidate, since the ATM cross-connects are designed to trap and transfer these cells to the control software efficiently. However, the limited cell size would preclude the insertion of information into the DRA messages that is not strictly needed for the DRA operation as such, but definitely is needed for *monitoring* the behaviour of the DRA for experimental purposes. For instance, several timestamps have to be inserted in the messages to estimate accurately the node model. For this reason, it has been decided to use AAL5 for the DRA messaging. Although this messaging system is less efficient due to the SAR functionality, it does not impose any limitations on the message size. The AAL5 layer has been integrated on the TLK, using the on-board cell drop functionality.
For the communication with the network manager, the AAL5 functionality foreseen for the DRA messaging is reused, and additionally the UDP/IP termination has been implemented. Alternatively, Ethernet/IP/UDP can be used.
Since the DRA has to be able to autonomously delete or create cross-connect points, and to detect alarms, the integration has to be such that at least these operations can be performed.

5.2 Location of the DRA in the control architecture

The location of the DRA should be such that speed of restoration is optimal. The speed of restoration is determined by the execution time of the core algorithm, and by the communication with the environment. The latter is determined by each of the components discussed above : communication between DRAs, and between DRA and the network manager, detecting failures, and setting up and tearing down connections. In order to minimise the total communication time, proper choices should be made for the integration of the DRA, by minimising the individual contribution of each of the components, and by paralleling the components as much as possible.
Some elements of the communication have to be considered to be fixed, e.g., the time needed for the detection and propagation of alarm through the control architecture is determined by the original architecture. Optimisation can only be done by putting the DRA as close as possible to the source of the alarm. The same holds for the creation/deletion of cross-connect points, and receiving/transmitting DRA messages. Thus, the DRA should be located as close to the links as possible, to minimise to message transfer delay within the control architecture.
However, the DRA has to modify routing tables and assign bandwidth to routes as an outcome of the algorithm. Since the DRA only has access to these functionalities if it is implemented on node level or higher, it has to be located at least at node level. Furthermore, the DRA should restore the transport service transparently for the TMN. Combining both requirements leads to the conclusion that the DRA should be located somewhere between the node connection module (refer to Figure 6) and the access to TMN.
The ideal location would thus be the node connection module. However, for the given ATM cross-connect, it is difficult to bring this choice in line with the objective to develop and validate the integration on the autonomous DRA with centralised network management. Indeed, the communication with the network manager, considered to be mandatory in this project, would be more easy if the DRA is integrated higher up, before the dispatching of the network management commands to the different applications. Therefore, it has been decided to integrate the DRA in the TMN-access module. Although this choice is, again, only sub-optimal in terms of core speed of restoration, it does allow to integrate network management in the testbed and the DRA.
Currently, the DRA has been integrated into the testbed environment and its robustness has been demonstrated for various network configurations of the testbed. The very first, and therefore non-optimised, results show that it is indeed possible to find the alternative routes within the call dropping time limit (± 2 seconds).

6 INTEGRATION OF NETWORK MANAGEMENT

Although the distributed restoration mechanism is operating fully autonomously, the network management station still has an important role to play in the whole process. Generally spoken, network restoration mechanisms can be described in 3 phases: pre-restoration, restoration and post-restoration.
During the pre-restoration phase the DRA has to be installed and initialised. Therefore, the network management station passes the relevant network-level information to the nodes (e.g.

the node-ids of the nearest neighbour nodes). Also the parameters of the DRA itself (e.g. the hop count) are initialised. In case more than one restoration mechanism is implemented in the network, the management station initialises the parameters that determine which mechanism will be activated under which circumstances.

During the restoration phase the DRA runs autonomously, and the network management station doesn't play any active role. However, a number of event reports can be generated by the DRA, which allows to monitor the restoration process on-line on the network management station. Also a logging mechanism has been implemented: the DRA can store the event information locally, and this data can be retrieved afterwards by the management station to obtain a complete restoration report, including timing information.

This brings us to the post-restoration phase, where the management station is responsible for collecting the event reports or log records and correlate this data into an overall restoration report, indicating the affected and restored connections, the restoration time, etc.

The implementation of these mechanisms requires adaptations both at the side of the management station (network management applications), and at the side of the DRA.

To allow DRA management data to be exchanged between the management station and the nodes, the MIB of the original ATM cross-connect is extended with a new DRA fragment. The mechanisms of on-line and off-line monitoring (respectively by event reports and log records) are mostly generic and can thus easily be adapted to incorporate also the management of other restoration algorithms.

Next to the management of the restoration system, of course, the necessary management applications are provided to set up SPCs over the network, and to monitor the state of the managed elements.

7 CONCLUSIONS : CURRENT STATUS AND FUTURE WORK

A testbed for experimenting with restoration in ATM networks has been realised. The testbed consists of five 8x8 STM1/ATM cross-connects, and is supervised by a central network management platform. Hereby, provisions have been made to incorporate a maximum of flexibility, in order to test various restoration mechanisms, while striving for optimal performance. Furthermore, the testbed results contain sufficient information to feedback the results of the tests into a simulation environment.

The distributed restoration algorithm, which has previously been reported in literature, has been integrated into the testbed environment, and its robustness has been demonstrated for various network configurations. Early results show that it is indeed possible to find the alternative routes within the call dropping time limit (2 seconds).

In a next stage, the project will extensively measure the contribution to the overall performance of the different time components of the processes the restoration consists of. These results will be used to define a validated representation of the node model behaviour, which can be fed back into the simulation environment. Then, the performance of the distributed restoration algorithm can be assessed for larger ATM networks. The testbed results will also result in requirements how to optimise the algorithm and its implementation into the ATM cross-connect.

The feasibility demonstration of other restoration mechanisms is also envisaged. For this, the necessary measures have been taken to provide the testbed with sufficient flexibility, such that other restoration schemes can be added in both the ATM cross-connect and the network management easily.

REFERENCES

Banniza, T.R., Eilenberger, G.J., Pauwels, B., Therasse, Y. (1991) Design and technology aspects of VLSI's for ATM switches. *IEEE Journal on Selected Areas in Communications*, **9**(8), 1255-1264

Bicknell, J., Chow, C.E., Syed, S. (1993) Performance analysis of fast distributed network restoration algorithms. *IEEE Globecom 93*, 1596-1600

Boettle, D., Henrion, M.A. (1990) Alcatel ATM switch fabric and its properties. *Electrical Communication*, **64** (2/3), 156-165

Chow, C.E., Bicknell, J., McCaughey, S. (1993a) A Fast Distributed Network Restoration Algorithm. *International Phoenix Conference on Computer and Communications*

Chow, C.E., McCaughey, S., Syed, S. (1993b) RREACT: A Distributed Network Restoration Protocol for Rapid Restoration of Active Communication Trunks. *Proc. of 2nd IEEE Network Management and Control Workshop*

Grover, W.D. (1987) The selfhealing network: a fast distributed restoration technique for networks using digital crossconnect machines. *IEEE Globecom 87*, 1090-5

Grover, W.D., Venables B.D., MacGregor M.H. and Sandham J.H. (1990) Development and performance assessment of a distributed asynchronous protocol for real-time network restoration. *IEEE Journal on Selected Areas in Communications*, **9**(1), 112-125

Han Yang, C., Hasegawa, S. (1988) FITNESS: Failure immunisation technology for network service survivability. *IEEE Globecom 88*, 1549-54

Kawamura, R., Sato, K., Tokizawa, I. (1992) Self-healing ATM network techniques utilizing Virtual Paths. *Networks 92*, 129-134

Kobrinski, H., Azuma, M. (1993) Distributed control algorithms for dynamic restoration in DCS mesh networks: performance evaluation. *IEEE Globecom 93*, 1584-8

Komine, H., Chujo, T., Ogura, T., Miyazaki, K., Soejima, T. (1990) A distributed restoration algorithm for multiple-link and node failures of transport networks. *IEEE Globecom 90*, 459-463

Nederlof, L., Struyve, K., O'Shea, C., Misser, H., Du, Y. and Tamayo, B. (1995a) End-to-end survivable broadband networks. *IEEE Communications Magazine*, **33**(9), 63-70

Nederlof, L., Vanderstraeten, H., Vankwikelberge, P. (1995b) A new distributed restoration algorithm to protect ATM meshed networks against link and node failures. *ISS 95*, A8.1

Vanderstraeten, H., Ocakoglu, G., Nederlof, L., Chantrain, D., Van Hauwermeiren, L. (1996) Simulation and implementation of a distributed algorithm for restoration in ATM meshed networks. *Computer Communications*, submitted

Wu, T.H. (1995) Emerging Technologies for Fiber Network Survivability. *IEEE Communications Magazine*, **33**(3), 58-73

Wu, T.H., Kobrinski, H. (1993) A service restoration time study for distributed control SONET digital cross-connect system self-healing networks. *ICC 93*, 893-9