Fault Tolerance on Grid-based ATM Switches

H.S. LASKARIDIS[1], A.A. VEGLIS[2], G.I. PAPADIMITRIOU[1] and A.S. POMBORTSIS[1]
[1]Dept. of Informatics, School of Science
[2]Computer Lab, School of Journalism & Mass Communication
Aristotle University of Thessaloniki
P.O. Box 888, 54006, Thessaloniki
GREECE

Abstract – As Asynchronous Transfer Mode technology becomes more widely used and consequently more critical, fault tolerance in ATM switches is considered to be of major importance, as it is quite unacceptable to have total disruption of operation due to failure in a part of the switch. In the present paper a new grid-based self-routing ATM switch fabric is presented, which exhibits great characteristics as far as fault tolerance is concerned, without employing redundancy. Fault tolerance is studied, by employing the metric of *"survival probability"* in order to evaluate the behavior of the architecture under concurrent failures.

Keywords - ATM switch fabric, grid-based, fault-tolerance, survival probability *CSCC'99* Proc.pp..2131-2137

1 Introduction

ATM is the high-edge technology employed in an increasing number of WAN production links, because of its high throughput capabilities. As ATM stops being an experimental network technology and becomes a production network technology, hardware fault tolerance becomes a crucial and important factor.

A lot of ATM switch architectures have been proposed in the literature. The large majority of these are highly modular and thus expandable. It is certainly unacceptable a failure in a small module to cause a total disruption of operation of the whole switch. Most of the architectures are variations of the well-known Banyan network topology, initially employed as an interconnection network in parallel systems. The majority of the variations employ redundant switching elements (SE's) and/or links (e.g. [1] - [6]), primarily used in order to achieve higher throughput. Such architectures are usually referred to as *"dilated"* interconnection networks and could also exhibit fault tolerant characteristics, if the proper logic was employed. However, fault tolerance is not described in these papers. In [7] - [9] fault tolerant switch architectures, also employing redundancy, were presented and analyzed.

In [10] a new non-Banyan-based switch fabric architecture, the *"Grid-based ATM Switch Architecture" (GASA)*, was presented. In the same paper performance issues (performance analytical model and simulation results) were presented and studied. In this paper the great fault tolerance characteristics exhibited by GASA without employing redundant SE's and/or links, are discussed. The rest of the paper is organized as follows: in section II the basic overall architecture is presented in summary. The enhancements, which are necessary in order to enable fault tolerance characteristics, are discussed in section III. In section IV the fault tolerance analysis is presented, using the metric of *"survival probability"*. Concluding remarks can be found in section V.

2 The Basic Overall Architecture

The basic structure of GASA consists of a grid of SE's. The number of SE's is equal to the number of input ports and the number of output ports. Each SE is directly connected to an input module, an output module and its 2 – 4 (depending on its position on the grid) neighbor SE's. Links between SE's are bi-directional while links between SE's and input or output modules are unidirectional. A 16×16 switch is presented in figure 1.

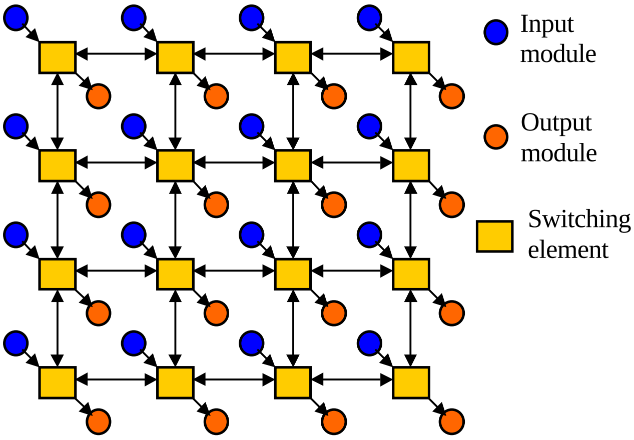
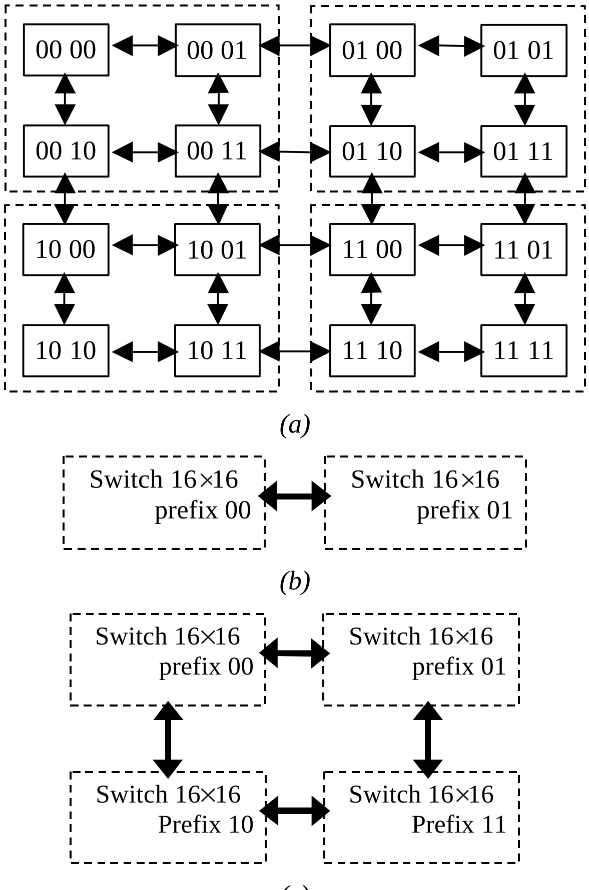


Fig. 1. A 16×16 GASA switch

Each SE has a SE identification number, from now on called *SE ID*. SE IDs are assigned in a recursive way as depicted in figure 2 (input and output modules are omitted for simplicity reasons).



(c)

Fig. 2. Switching element addressing scheme
(a) 16×16 switch (b) creating a non-square 32×32 switch
(c) creating a square 64×64 switch

Having assigned the SE IDs in that particular manner, routing of cells can be accomplished by the following algorithm, executed in each SE, independently of its position on the grid. The algorithm presented is used for routing in a 16×16 switch, but it is easily scalable to any dimension. It is worth noting that no central routing control is deployed, as it is considered to be a bottleneck. In this algorithm, `SE_ID[a]` denotes the ath bit of the SE ID of the current SE, `SE_ID[a,b]` denotes the ath and bth bits of the SE ID, and `Dest_SE_ID[a]` denotes the ath bit of the SE ID of the destination SE.

```
Normal Routing
If Dest_SE_ID[1,2] <> SE_ID[1,2] then
   Route on (1,2)
Else
   If Dest_SE_ID[3,4] <> SE_ID[3,4] then
      Route on (3,4)
   Else
      Send to output module
```

```
Procedure Route on (a,b)
If Dest_SE_ID[a] <> SE_ID[a] then
   If Dest_SE_ID[a] = 0 then
      Send to North
   Else{Dest_addr[a]=1}
      Send to South
Else {Dest_SE_ID[a]= SE_ID[a],but}
      {Dest_SE_ID[b]<>SE_ID[b]    }
   If Dest_SE_ID[b] = 0 then
      Send to West
   Else
      Send to East
```

Regarding the SE architecture, a shared queue architecture was deployed, which can also support multiple-level priorities. The reader should refer to [10] for more details.

3 Fault tolerant Enhanced Architecture

In order to enable fault tolerant capabilities, certain modifications have to be made:

1. There should be a central unit controlling and resolving faults, from now on called *“Fault Recovery Control Unit" (FRCU).*
2. There should be a “communication network” connecting each SE to the FRCU. A number of alternatives will be presented, along with their advantages and disadvantages.
3. There should be hardware in each port of each SE able to identify and report faulty links.
4. The routing algorithm should be adjusted, in order to be able to operate under a slightly different way, when FRCU announces a fault.

3.1 The fault recovery communication network

The *“fault recovery subsystem”* consists of the *“Fault Recovery Control Unit" (FRCU)* and the communication network connecting FRCU to each SE and vice versa. The operation of FRCU is presented in the subsequent subsection, while in the present subsection the alternatives that can be deployed for the communication network are presented.

The possible alternatives (illustrated in fig. 3) are the following:

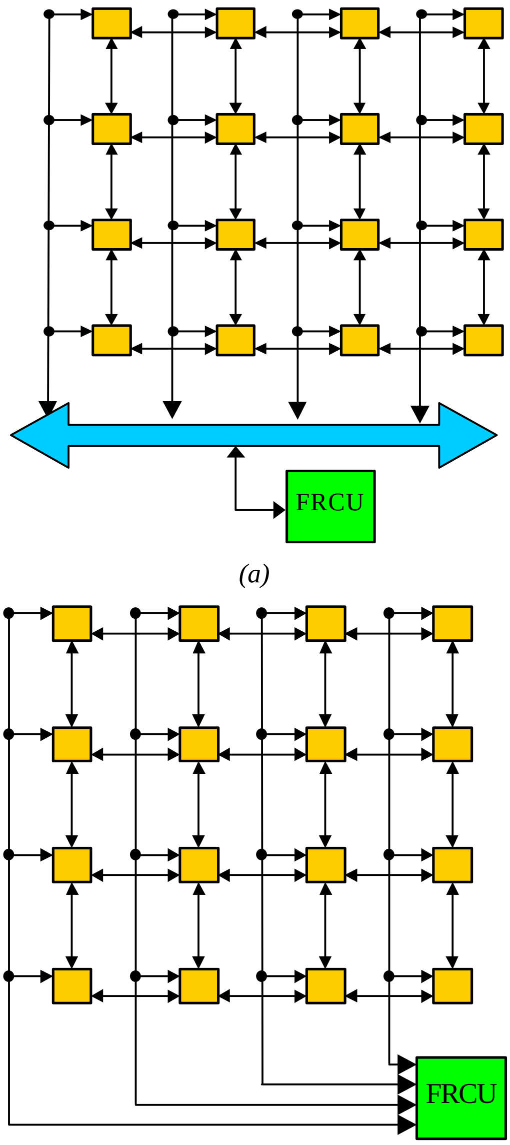
1. *Use of a shared bus:* All SE’s as well as the FRCU are connected to a shared bus. The advantage of this approach is low cost, while on the other hand the shared bus is a “single point of failure”.
2. *Use dedicated lines between each SE and the FRCU:* Each SE is directly connected to the FRCU by a dedicated bi-directional link. This

approach provides maximum fault tolerance, while its disadvantages are the high cost and its restricted scalability, due to the fact that the FRCU should have as many ports as the number of SE's in the switch.

3. *Use of multiple buses:* Each row of SE's has its own bus. FRCU is connected to all buses. This solution is considered to be a compromise between the above mentioned approaches.

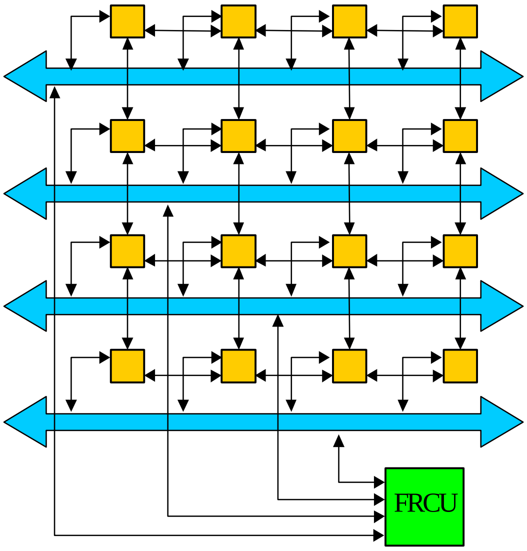
The bandwidth of the buses used in alternatives 1 and 3 are not of major importance, as the traffic on these buses is minimal.

It is worth noting that failure in any component of the communication network (no matter which approach is used) results in fault recovery scheme not being totally operational, but does not result in any kind of disruption of the switch's basic operation.



(a)

(b)



(c)

Fig. 3. The fault recovery communication network, connecting the FRCU to SE's

Because of this fact, in section IV, links and buses used in the fault recovery communication network are not taken under consideration during fault tolerance analysis.

3.2 The operation of FRCU

In the present subsection the operation of the FRCU is discussed using pseudocode. The operation of the Fault Recovery Subsystem depends on a communication protocol consisted of 3 control messages exchanged between the FRCU and SE's:

1. *Notif_Link_failure (SE_ID, Direction)* is a notification message sent by a SE to the FRCU when the SE discovers that one of its links has gone down.
2. *Announc_SE_failure (SE_ID)* is a broadcast announcement sent by the FRCU to all SE's when the FRCU concludes, by considering the combination of received faulty links reports, that there is a faulty SE.
3. *Announc_Link_failure (SE_ID)* is a broadcast announcement sent by the FRCU to all SE's when a fault on the link connecting a SE to its corresponding output module is reported by the SE.

The purpose of the last two control messages is that on receipt of such messages, all SE's can drop any cells destined to the specified SE, as there is no reason forwarding them; such cells will never manage to reach their destination, as their destination is not operational or unreachable.

The FRCU has to be aware of the grid topology. For this purpose an adjacency matrix

(Adj_Matrix) is stored in the FRCU. A list of received reports (Reports_List) is also maintained in the FRCU, sorted by SE ID. The operation of the FRCU can be described by the following pseudocode.

```
On receipt of msg Notif_Link_failure (SE_ID, Direction)
   If SE_ID ∈ Probable_faulty_SE then
      Remove SE_ID from Probable_faulty_SEs
   If Direction = Output_module then
      Send msg Announc_Link_failure (SE_ID)
   Else
      Corresponding_SE := Adj_Matrix[SE_ID, Direction]
      Add new record [SE_ID, Direction] in Reports_List
      Add Corresponding_SE in Probable_faulty_SEs
      If ∃ reports in Reports_list from
      all neighbors of Corresponding_SE ∉ Probable_faulty_SEs then
         Send msg Announc_SE_failure (SE_ID)
```

3.3 The operation of each SE

Each SE preserves a list of SE's announced to be faulty or having faulty link connecting them to the corresponding output module (Faulty_SEs_set). The pseudocode in fig. 4 presents the basic operation of each SE, as far as fault recovery is concerned.

In order to overcome the problem of faulty links connecting operational SE's, care must be taken so that cells misrouted on purpose to another SE do not return to the transmitting SE. For this purpose, an additional bit is used in each cell 's tag, from now on called *"NR"* (standing for Normal Routing). Value "0" stands for normal routing, while value "1" stands for inverse routing, i.e. routing done by firstly considering the 2nd bit of each pair of bits.

The routing algorithm presented in fig. 5 should be employed.

```
On link going down
   Send msg Notif_Link_failure (my_SE_ID, Direction)
On receipt of msg Announc_Link_failure (SE_ID)
   Add SE_ID to Faulty_SEs_set
On receipt of msg Announc_SE_failure (SE_ID)
   Add SE_ID to Faulty_SEs_set
```

Fig. 4. Basic operation of a SE, regarding fault recovery

```
Procedure Routing
If Dest_SE_ID[1,2] <> SE_ID[1,2] then
   Route on (1,2)
Else
   If Dest_SE_ID[3,4] <> SE_ID[3,4] then
      Route on (3,4)
   Else
      Send to output module

Procedure Route on (a,b)
If NR = 0 then
   If Dest_SE_ID[a] <> SE_ID[a] then
      If Dest_SE_ID[a] = 0 then
         Send to North
      Else {Dest_addr[a]=1}
         Send to South
   Else {Dest_SE_ID[a]=SE_ID[a],but Dest_SE_ID[b]<>SE_ID[b]}
      If Dest_SE_ID[b] = 0 then
         Send to West
      Else
         Send to East
Else {NR = 1}
   Inverse Route on (a,b)
```

```
Procedure Inverse Route on (a,b)
If Dest_SE_ID[b] <> SE_ID[b] then
   If Dest_SE_ID[b] = 0 then
      Send to West
   Else
      Send to East
Else {Dest_SE_ID[b]=SE_ID[b],but Dest_SE_ID[a]<>SE_ID[a]}
   If Dest_SE_ID[a] = 0 then
      Send to North
   Else {Dest_addr[a]=1}
      Send to South

Procedure Inverse Routing
If Dest_SE_ID[1,2] <> SE_ID[1,2] then
   Inverse Route on (1,2)
Else
   If Dest_SE_ID[3,4] <> SE_ID[3,4] then
      Inverse Route on (3,4)
   Else
      Send to output module

Procedure Check
If cell destined to SE ∈ Faulty_SEs_set then
  Discard cell
Else
   If cell destined to operational link then
      Send cell to that link
   Else
      If cell destined to faulty North/South link then
         Set NR := 0
         Reverse Routing
         Check
      If cell destined to faulty West/East link then
         If both North/South links are down then
            If both West/East links are down then
               Discard cell
            Else
               Set NR := 0
               Send to operational West/East link
         Else
            Set NR := 1
            Send to operational North/South link
      If cell destined to faulty output-module link then
         Discard cell
```

Fig. 5. The fault tolerant routing algorithm executed in each SE

The possible “paths” between procedures that a cell can follow in a SE are depicted in figure 6.

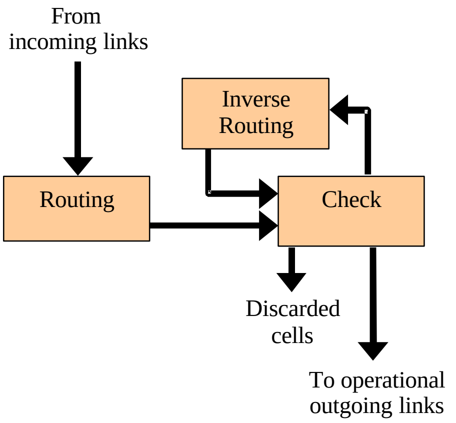


Fig. 6. Possible “paths” of a cell in a SE

It should be noted that the FRCU is a “single point of failure”. If both the FRCU and a SE fail, then the neighbor SE’s will consider that there are only faulty links (not faulty SE), given that they don’t receive a relevant announcement from the FRCU. That would lead to cells being transferred from one SE to another forever, and subsequently lead to buffers’ overflow. So we need a way that cells destined to faulty SE’s would be discarded even though the FRCU has failed. For this purpose we can use “hop-counters”. Each cell has a hop-counter in its tag, which is checked on entering each SE. If the hop-counter is greater than a threshold it is discarded, otherwise it is increased by one. The threshold should be slightly higher than the diameter of the grid, so that diversions due to faulty links would not lead to discarding cells destined to operational reachable SE's.

4 Fault Tolerance Analysis

In order to estimate the fault tolerance characteristics of GASA switches we adopt the notion of *"Survival probability function"*, introduced by A. Itoh in [9]. We consider all failures as random independent incidents and we define:

- Link Survival probability $S_{link}(k)$ as the probability that all SE's of the switch can communicate with each other, although there are k faulty links.
- SE Survival probability $S_{SE}(k)$ as the probability that all operational SE's of the switch can communicate with each other, although there are k faulty SE's.

In order to calculate the values of the functions, the following equations are used:

$$S_{link}(k) = \frac{\# \mathrm{DoC}}{\binom{\# \mathrm{links}}{k}} \quad (1)$$

$$S_{SE}(k) = \frac{\# \mathrm{DoC}}{\binom{\# \mathrm{SE's}}{k}} \quad (2)$$

where DoC is the number of cases that "disruption of communication" occurred. The switch fabric is considered disrupted when there is at least one pair of input and output modules, connected to operational SE's, that cannot communicate through the fabric, as there is no path connecting these modules.

In equation 1, we consider all combinations of k faulty links that lead to disruption of communication between SE's, divided by the total number of combinations of k links. It should be noted that only links connecting SE's were taken into account, while links connecting SE's to input and output modules were not taken into account, as they can never lead to disruption of communication between SE's. Similarly in equation 2, we consider all combinations of k faulty SE's that lead to disruption of communication between the rest (operational) SE's, divided by the total number of combinations of k SE's.

In fig. 7 results from the link Survival Probability function are presented for a 16×16 GASA switch. In fig. 8 results from the SE Survival Probability function are presented for both 16×16 and 32×32 GASA switches. In the x-axis the *percentage* of faulty links or SE's is shown, in order to make comparisons easier.

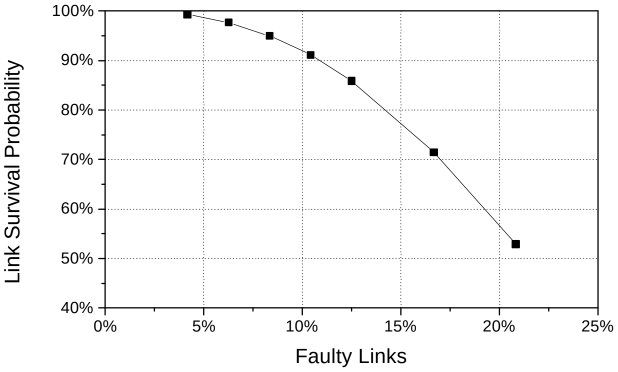


Fig. 7. Link survival probability for a 16×16 GASA

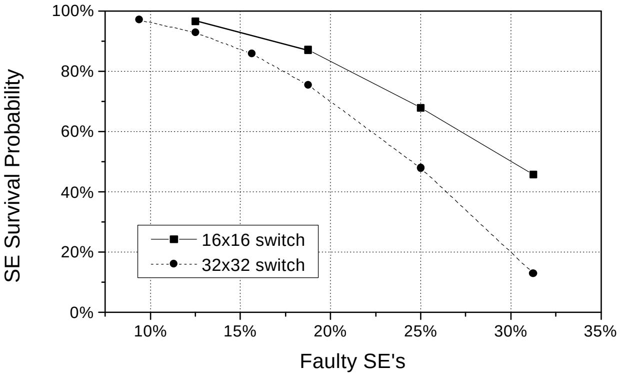


Fig. 8. SE survival probability for 16×16 and 32×32 GASA

From the above results, it can be noticed that the GASA switch has a remarkable behavior regarding fault tolerance: even when 20% of the links have failed, the probability that all SE's can communicate with each other is grater than 0.5. Even when 25% of the SE's have failed, the probability that the operational SE's can communicate with each other is over 0.65 (in 16×16 switch) and over 0.45 (in 32×32 switch).

5 Conclusions

In [10] a new self-routing grid-based ATM switch architecture was presented. In this paper we focused on the fault tolerance characteristics that the switch fabric exhibits without employing redundant SE's or redundant links. The necessary additions to the basic architecture, the enhanced operation of the SE's and the operation of the Fault Recovery Control Unit (FRCU) were presented. A simple protocol used in the communication between the FRCU and the SE's was also defined.

In order to evaluate the fault tolerance, the notion of survival probability was used. Results were calculated and presented which demonstrate in a graphical way the behavior of the GASA switch under concurrent failures.

GASA is easily extensible to 3-dimentional grid. We strongly believe that such a 3-dimentional architecture would exhibit even better characteristics, regarding fault tolerance, as it would form a more “strongly connected” graph. A future work will include the evaluation of the 3-D architecture, using the same metric, survival probability.

References:

[1] F.A. Tobagi, T. Kwok, F.M. Chiussi, “Architecture, performance, and implementation of the Tandem Banyan fast packet switch”, *IEEE Journal on Sel. Areas in Communication*, Vol. 9, No. 8, 1991, pp. 1173-1193

[2] J.N. Giacopelli, J.J. Hickey, W.S. Marcus, W.D. Sincoskie, M. Littlewood, “Sunshine: a high-performance self-routing broadband packet switch architecure”, *IEEE Journal on Sel. Areas in Communication*, Vol. 9, No. 8, 1991, pp. 1289-1298

[3] T.T. Lee, S.C. Liew, “Broadband packet switches based on dilated inteconnection networks”, *IEEE Transactions on Communications*, Vol. 42, No. 2/3/4, 1994, pp. 732-744

[4] I. Widjaja, A. Leon-Garcia, “The Helical-Switch: A multipath ATM switch which preserves cell sequence”, *IEEE Transactions on Communications*, Vol. 42, No. 8, 1994, pp. 2618-2629

[5] S.F. Oktug, M.U. Caglayan, “Design and performance evaluation of a Banyan network based interconnection structure for ATM switches”, *IEEE Journal on Sel. Areas in Communication*, Vol. 15, No. 5, 1997, pp. 807-816

[6] M. Collier, “A three-stage ATM switch with cell-level path allocation”, *IEEE Transactions on Communications*, Vol. 45, No. 6, 1997, pp. 701-709

[7] T. Zhang, A.K. Somani, “DIRSMIN: A fault-tolerant switch for B-ISDN applications using dilated reduced-stage MIN”, *Proc. of INFOCOM ’95*, 1995, pp. 643-650

[8] S.C. Yang, J.A. Silvester, “A reconfigurable ATM switch fabric for fault tolerance and traffic balancing”, *IEEE Journal on Sel. Areas in Communication*, Vol. 9, No. 8, 1991, pp. 1205-1217

[9] A. Itoh, “A fault tolerant switching network for B-ISDN”, *IEEE Journal on Sel. Areas in Communication*, Vol. 9, No. 8, 1991, pp. 1218-1226

[10] H.S. Laskaridis, A.A. Veglis, G.I. Papadimitriou, A.S. Pomportsis, “Grid-based ATM Switch Architecture: a new fault-tolerant space-division switch fabric architecture”, to be published in *Proc. of IEEE Mediterranean ’99*, 1999