Performance Analysis of a Fault-tolerant ATM Switch in Single-fault Conditions

S. Segkhoonthod
M. C. Sinclair
Dept. of Electronic Systems Engineering, University of Essex,
Wivenhoe Park, Colchester, Essex C04 3SQ.
Tel: 01206-872477; Fax: 01206-872900; email: {segks,mcs}@essex.ac.uk

1 Introduction

One of the approaches for ATM switch design is to use a number of routing networks (multistage interconnection networks: MINs) in parallel so that multiple paths are created between inputs and outputs [1, 2]. This technique, called a replicated network, results in an improvement in both performance and reliability. However, it should be noticed that with this configuration, a great deal of complexity is needed in the Input Modules (IMs), which are used to distribute packets among routing networks, as they have to decide to which routing network packets should be sent.

In [3], we proposed a fault-tolerant ATM switch which adapts the idea of using several routing networks in parallel. The switch has a distribution network which is in charge of dispatching packets among the routing networks. As a result, the IMs are simply required to submit packets to the distribution network. Since the distribution network also carries out part of routing process, the routing networks themselves are truncated—resulting in a low overall complexity for the switch. In terms of fault tolerance, the distribution network is also in charge of re-routing packets in the case of faulty switching elements or links. Unlike some other fault-tolerant switches, where additional effort is required to avoid faults, in the proposed switch there is no modification in the routing algorithm when faults occur. In [3], we presented a probabilistic analysis of the proposed switch with re-routing deactivated, as well as simulation results for re-routing in the absence of faults. It was shown that the proposed switch gives both improved performance and reduced complexity when compared with replicated networks. In [4], we presented a performance analysis of the switch with re-routing in the absence of faults, based on the method of state diagrams, and compared the results with simulation. The results obtained showed reasonable agreement between the analysis and simulation. Further, the performance of the proposed switch with re-routing is significantly better than without. In this paper, we extend the work in [4] by taking into account the presence of a single fault in various parts of the switch.

2 Switch Architecture and Operation

An l-level $N \times N$ version of the proposed switch consists of an $N \times K$ distribution network (DN), K $N \times N$ routing networks (RNs) and N $K \times 1$ output modules (OMs) where $K = 2^l$. Shown in Fig. 1 is an $l = 2$, $K = 4$, 16×16 version of the proposed switch. Basically, the structure of the proposed switch is derived by interleaving the first few stages of K Banyan networks into a tree configuration by means of 2×4 switching elements (SEs). For example, in Fig. 1, there are 4 Banyan networks embedded with their stages 0 and 1 overlapped in the DN. The DN is constructed based on an l-level binary tree where each node is a plane of $N/2$ 2×4 SEs. In stage s, there will be 2^s such planes. The 2×4 SE used in the DN is not a typical 2×4 crossbar; it behaves like a 2×2 SE, but with two identical sets of two outputs, instead of only one set as in case of a 2×2 SE. Given an l-level DN, there will be $2^l = K$ RNs. Each output of the RNs is connected to one of the inputs of an OM. Here an OM can accept K packets at the same time.

The DN and RNs are both unbuffered. The switch operation is based on a request/acknowledgement protocol, consisting of two phases: path-establish and packet-transfer [5]. At the beginning of the first phase, an IM that has a packet generates a request, a form of short packet, and then submits it to the DN and waits for an acknowledgement (ACK). If the returned ACK is a positive one, the IM can transfer the packet to the appropriate OM during the second phase. If not, it waits for the next path-establish phase and starts the same procedure again. The DN is in charge of distributing, routing and re-routing. Basically, it both distributes an incoming request to one of the RNs according to some rule, and at the same time routes the request. After receiving a request from the DN, the RN then directs it to an appropriate OM. The routing algorithm used is the same as that of a Banyan network. The various options for the operation of the 2×4 SEs are given in [4]; here we consider the switch in re-routing mode with random selection. The 2×4 SEs employed in the DN have two sets of outputs, referred to as first and second choices. In each set, there are two outputs—upper and lower outputs like those of a 2×2 SE. In random selection, if a request is addressed to the upper output of a 2×4 SE, it is randomly granted one of the choices. If the request can not get through to the next stage via that choice, an attempt will be made to re-route it using the other choice—hence, re-routing mode.

3 Performance Evaluation

3.1 Assumptions and Fault Model

Before the analysis, a few definitions are required. A *stage cycle* is the time interval during which a request (or an acknowledgement) moves from one stage to an adjacent stage. A *network cycle* is the time interval for a packet received at an IM to pass through to an OM. In terms of the switch's operation described above, one network cycle then consists of the sum of the path-establish and packet-transfer phases. Further, we assume here that a packet may only arrive at an IM at the beginning of a network cycle. In which case, one network cycle consists of a finite number of stage cycles[1]. Another assumption, which applies to the situation when a 2×4 SE performs re-routing, is that it is assumed that re-routing cannot take place during the same cycle. This assumption, which was also adopted in [5], is needed to simplify the analysis, although in practice a request may be able to try an alternative path within one stage cycle. The fault model adopted is failure of an individual SE, leaving it totally unusable [6]. We will focus on a single SE fault in a given stage of the switch. Further, we will assume that the SEs in stage 0, the IMs and the OMs are all fault-free, as the failure of such components could lead to the failure of the whole switch. In addition, when a request encounters a faulty SE, we will assume that the faulty SE behaves as if it was always occupied; a request is therefore always blocked. In other words, a fault and blocking are treated identically. We analyse the proposed switch using the same assumptions as in [4]: independent, uniform traffic and infinite buffering at an OM. In addition, we assume that a fault occurs randomly. The performance measure used is the *probability of acceptance* (P_{A}) which is defined as the probability that an arbitrary request will be accepted [8, 9].

3.2 State Diagram Technique

In the method of state diagrams, two things are required [10]: (1) a finite number of states in a diagram and (2) transition probabilities between states. In the state diagram, the future state of a system depends only on its current state. Once the above two parts are completed, one can then obtain the probability of being in any desired state. To apply the method of state diagrams to the analysis of a re-routing version of the proposed switch, it is appropriate that the state diagram for the analysis be a diagram whose states are the "state" of a request in a particular stage (in a particular cycle) where the future state of the request (in the next cycle) only depends on its current state and the current state of the switch. So the task here is to first build a diagram with appropriate states, and then work out transitions probabilities among the states. Such a state diagram for an $l = 2$, $K = 4$, 16×16 version of the proposed switch is depicted in Fig. 2.

3.3 State Description

In Fig. 2, the states are identified by several components as follows:

- **Stage Number** (s) indicates the stage number where the request is.
- **Path Number** (p) indicates the path number a request is using. Recall that in the proposed switch, there exist 2^l paths (not disjoint) between every IM and OM pair. Here path number 1 is defined as the first path

[1]Hereafter, the term "cycle" refers to the stage cycle, unless otherwise stated.

that a request takes in attempting to go its required OM, path number 2 is the second one and so on.

- **Cycle Number** (c) indicates the cycle number that a request is in.
- **Blocking Number** (b) indicates the stage number where a request has been most recently blocked.

Consider an l-level $N \times N$ switch with $n = \log_2 N$ stages and $K = 2^l$. By means of the above components, the state of a request can be uniquely identified and classified into four groups:

Set-up: Being in a Set-up state means that a request has so far been successful in establishing a path up to the stage where it is currently, and it will be trying to establish a path to the next stage during the next cycle. In terms of the physical switch, this refers to when a request is allowed to occupy an output of an SE in a given stage. For example, state 1 in Fig. 2 may be interpreted as a request now being at one of the outputs of an SE in stage 0. Another example is state 7, which is a scenario when, after having been blocked previously, the request is able to be re-routed (*i.e.* set up). Set-up states can be further divided into three types as follows:

S_0 Request is at the IM and is being submitted to the first stage of the DN in cycle 1. This is state 0 in Fig. 2.

$S_{s,p,c}$ Request is in stage s, it is currently using path number p, it has not been blocked before, and it will be attempting to set up a path to stage $s + 1$ in cycle $c + 1$. These are states 1, 2, 3 and 4 in Fig. 2.

$S^b_{s,p,c}$ Request is in stage s, it is currently using path number p, it has been blocked most recently at stage b and it will be attempting to set up a path to stage $s + 1$ in cycle $c + 1$. Examples include states 7, 8 and 9.

Blocked: Being in a Blocked state has one of three meanings. The first refers to a situation when a request in stage s gets blocked when trying to establish a path to stage $s+1$; a negative acknowledgement is received at stage s. From Fig. 2, states 6, 14 and 18 belong to this category. The second is when a request, after having been blocked previously, is blocked again as a result of the other output in a 2×4 SE being unavailable in an attempt to re-route the request. Examples include states 12 and 26. The third is when a request has already been blocked at some stage ahead of where it currently is and it is on the way back to some stage in the DN from where it may be re-routed. These include states 30 and 33. A Blocked state is defined as below:

$B^b_{s,p,c}$ Request is in stage s during cycle c, it was blocked at stage b when attempting to set up a path to stage $b + 1$ with path number p.

Hold: Being in a Hold state means that the request is at its required OM and a path is being held between its IM and OM for a number of cycles before the packet-transfer phase starts. Once it is in a Hold state, it will be in that state until the end of the path-establish phase. The Hold states are divided into two types:

$H_{p,c}$ Request has been successful in setting up a path from its IM to its requested OM by using path number p, and the path is being held between this source-destination. This is state 5 in Fig. 2.

$H_{p,c}^{b}$ Request is in its required OM after having been successful in setting up a path from its IM to the OM by using path number p, it has been blocked most recently at stage b and the path is being held between this source-destination. Examples include states 10, 24, 39 and 44.

Lost: When a request is in a Lost state, it is back at its originating IM after having failed on all paths. Once in a Lost state, it will be there until the last stage cycle of the path-establish phase. A Lost state is defined as below:

$L_{p,c}^{b}$ Request is at its originating IM after being unable to set up a path, the most recent path number used was p and it was most recently blocked at stage b. Examples include states 15 and 19.

3.4 Transition Probabilities Derivation

Denote the probability that a request will be in state A by $P[A]$ and the transition probability from state B to state A by $P[A|B]$ where A and B are any of the states defined in the previous section and A is not the same state as B. The transition probabilities of each state are given and described below:

3.4.1 Request is currently in a Set-up state

$$P[S_{0,1,1}|S_0] = 1 \quad \text{(At an IM)} \tag{1}$$

(1) Request has not been blocked before

With a fault in stage f, $1 \leq f \leq l-1$:

$$\left.\begin{aligned} P[S_{s+1,1,c+1}|S_{s,1,c}] &= \frac{(2^{s+1})(\frac{N}{2})-1}{(2^{s+1})(\frac{N}{2})} \\ P[B^{s}_{s,1,c+1}|S_{s,1,c}] &= 1 - P[S_{s+1,1,c+1}|S_{s,1,c}] \end{aligned}\right\} (s = f - 1) \tag{2}$$

$$P[S_{s+1,1,c+1}|S_{s,1,c}] = 1 \quad (0 \leq s \leq l-2,\ s \neq f-1) \tag{3}$$

Otherwise:

$$P[S_{s+1,1,c+1}|S_{s,1,c}] = 1 \quad (0 \leq s \leq l-2) \tag{4}$$

With a fault in stage f, $l \leq f \leq n-1$:

$$\left.\begin{aligned} P[S_{s+1,1,c+1}|S_{s,1,c}] &= \left(\frac{\frac{KN}{2}-1}{\frac{KN}{2}}\right) \times \\ \Big((1 - P[S_{s,1,c}]/K) &+ 0.75(P[S_{s,1,c}]/K)\Big) \\ P[B^{s}_{s,1,c+1}|S_{s,1,c}] &= 1 - P[S_{s+1,1,c+1}|S_{s,1,c}] \end{aligned}\right\} (s = f - 1) \tag{5}$$

$$\left.\begin{aligned} P[S_{s+1,1,c+1}|S_{s,1,c}] &= (1 - P[S_{s,1,c}]/K) + 0.75(P[S_{s,1,c}]/K) \\ P[B^s_{s,1,c+1}|S_{s,1,c}] &= 1 - P[S_{s+1,1,c+1}|S_{s,1,c}] \end{aligned}\right\} \quad (6)$$

where $l - 1 \leq s \leq n - 2$ and $s \neq f$-1.

Otherwise:

$$\left.\begin{aligned} P[S_{s+1,1,c+1}|S_{s,1,c}] &= (1 - P[S_{s,1,c}]/K) + 0.75(P[S_{s,1,c}]/K) \\ P[B^s_{s,1,c+1}|S_{s,1,c}] &= 1 - P[S_{s+1,1,c+1}|S_{s,1,c}] \end{aligned}\right\} \quad (7)$$

where $l - 1 \leq s \leq n - 2$.

$$P[H_{1,c+1}|S_{n-1,1,c}] = 1 \quad (s = n - 1) \quad (8)$$

We will consider the fault-free case first. Eq. 1 states that whenever there is a request, the IM will always submit it to the first stage of the DN. $P[S_0]$ is also the input rate of the switch. Eq. 4 is derived from the fact that a request is always successful in establishing a path to a 2×4 SE in a stage of the DN because there are at most two requests coming to the inputs but there are four outputs available so there is always a free output. Eq. 7 comes from the fact that when a request is in stage s and wishes to set up a path via a 2×2 SE in stage $s+1$, it can do so if firstly there is no other request that will be arriving at this particular SE (the first term), or secondly there is another request that will arrive at the SE, but it will want the other output of the SE that is not wanted by the considered request (the second term). The probability that there is another request arriving at the other input of the 2×2 SE under consideration is expressed as $P[S_{s,1,c}]/K$ since the random selection assumption is adopted. Eq. 8 means that a request in stage $n-1$ can always set up a path to its addressed OM as a result of the multiple acceptance assumption. This is state 4 of Fig. 2. In the case where a fault is present in a given stage, the probability of a request getting through that stage now includes the probability that the request is routed to the faulty SE. If we let M be the number of SEs in a stage where a faulty SE is, then the probability that a request is not routed to the faulty SE is $\frac{M-1}{M}$ given that a fault occurs randomly. So Eq. 2 and 5 both follow the above argument. As an example, Fig. 3 shows a diagram of an l=2, K=4, 16×16 switch proposed switch with a fault in stage 1. Suppose a request is in stage 0, state 1; thus in the next cycle it is going to attempt to set up a path to stage 1. However, one of the SEs in stage 1 is unusable, and it may be the SE that this request requires. In which case, the request will be treated as blocked and will stay at the same stage, state 105. Otherwise, it can proceed to stage 1, state 2.

(2) Request has been blocked before

In order to derive transition probabilities for states that have been blocked before, the following terms are defined:

$$\mathrm{F1}(s,c) = \sum_{i=s}^{n-1}\sum_{p=1}^{K}\sum_{b=0}^{n-2}\left(P[S^b_{i,p,c}] + P[S_{i,p,c}] + P[B^b_{i,p,c}] + P[H_{p,c}] + P[H^b_{p,c}]\right) \quad (9)$$

$$\mathrm{F2}(s,c) = \sum_{p=1}^{K}\sum_{b=0}^{n-2} P[S^b_{s,p,c}] \tag{10}$$

Eq. 9 gives the probability that in cycle c, there is a request which may be in one of the stages starting from stage s and beyond (including the OM). In other words, it gives the probability that there are other requests ahead of the considered one. As in Fig. 2, suppose that we are considering state 7, then states that are ahead of it during the same cycle are states 4 and 18, *e.g.* $\mathrm{F1}(2,4) = \mathrm{P}[\mathrm{S}_{3,1,4}] + \mathrm{P}[\mathrm{B}^2_{2,1,4}]$. Eq. 10 gives the probability that, in stage cycle c, requests in stage s will be setting up a path to stage $s+1$ during the next cycle.

With a fault in stage f, $1 \le f \le l-1$,

a) a request has already encountered the fault:

$$P[S^b_{s+1,p,c+1}|S^b_{s,p,c}] = 1 \quad (s = f-1) \tag{11}$$

b) a request has not yet encountered the fault:

$$\left.\begin{aligned} P[S^b_{s+1,1,c+1}|S^b_{s,1,c}] &= \frac{(2^{s+1})(\frac{N}{2})-1}{(2^{s+1})(\frac{N}{2})} \\ P[B^s_{s,1,c+1}|S_{s,1,c}] &= 1 - P[S_{s+1,1,c+1}|S_{s,1,c}] \end{aligned}\right\} (s = f-1) \tag{12}$$

$$P[S^b_{s+1,p,c+1}|S^b_{s,p,c}] = 1 \quad (0 \le s \le l-2,\ s \ne f-1) \tag{13}$$

Otherwise:

$$P[S^b_{s+1,p,c+1}|S^b_{s,p,c}] = 1 \quad (0 \le s \le l-2) \tag{14}$$

Again, we consider the fault-free case first. Eq. 14 states that when a request is blocked, and then is in a Set-up state again in a stage in the DN and is attempting to set up a path to the next stage, it will always be successful in doing so. Considering a request which is now in a Set-up state in stage s, this means that the request is at the output of a 2×4 SE in that stage and is going to one of the inputs of a 2×4 SE in stage $s+1$. Since there is at most one other request in the SE in stage $s+1$ that can be in competition, the request always gets through stage $s+1$. State 35 of Fig. 2 is an example of such a state. With a fault, however Eq. 12 follows the same argument as above, where the probability that a request is not routed to the faulty SE is $(2^{s+1}\frac{N}{2}-1)/(2^{s+1}\frac{N}{2})$. Eq. 11 follows the same argument as Eq. 14 as a request has already encountered a fault. State 106 of Fig. 3 is an example of such a state.

With a fault in stage f, $l \leq f \leq n-1$:

$$\left.\begin{array}{rcl} P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] & = & \\ \left(\frac{\frac{KN}{2}-1}{\frac{KN}{2}}\right) & \times & \Big\{0.75\big(P[S_{s,p,c}]+\mathrm{F2}(s,c)\big)/K \\ & + & 0.5\big(\mathrm{F1}(s+1,c)\big)/K \\ + \quad \big(1-P[S_{s,p,c}]/K & - & \mathrm{F2}(s,c)/K-\mathrm{F1}(s+1,c)/K\big)\Big\} \\ P[B^{s}_{s,p,c+1}|S^{b}_{s,p,c}] & = & 1-P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] \end{array}\right\} \quad (15)$$

where $s = f-1$.

$$\left.\begin{array}{rcl} P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] & = & 0.75\big(P[S_{s,p,c}]+\mathrm{F2}(s,c)\big)/K \\ & + & 0.5\big(\mathrm{F1}(s+1,c)\big)/K \\ + \quad \big(1-P[S_{s,p,c}]/K & - & \mathrm{F2}(s,c)/K-\mathrm{F1}(s+1,c)/K\big) \\ P[B^{s}_{s,p,c+1}|S^{b}_{s,p,c}] & = & 1-P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] \end{array}\right\} \quad (16)$$

where $l-1 \leq s \leq n-2$, $s \neq f-1$.

Otherwise:

$$\left.\begin{array}{rcl} P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] & = & 0.75\big(P[S_{s,p,c}]+\mathrm{F2}(s,c)\big)/K \\ & + & 0.5\big(\mathrm{F1}(s+1,c)\big)/K \\ + \quad \big(1-P[S_{s,p,c}]/K & - & \mathrm{F2}(s,c)/K-\mathrm{F1}(s+1,c)/K\big) \\ P[B^{s}_{s,p,c+1}|S^{b}_{s,p,c}] & = & 1-P[S^{b}_{s+1,p,c+1}|S^{b}_{s,p,c}] \end{array}\right\} \quad (17)$$

where $l-1 \leq s \leq n-2$.

We explain the fault-free case first. Eq. 17 is the probability that the request is successful in setting up a path to a 2×2 SE in the next stage. In attempting to set up a path to the next stage, the request has to encounter one of the three situations depicted in Fig. 4. Case (c) will occur when there is another request in the same stage which is also going to set up a path in the same cycle. This is expressed as $(P[S_{s,p,c}]+\mathrm{F2}(s,c))/K$. The probability that case (b) will occur is the probability that there is another request in some stage ahead of the considered one during the same cycle, and is expressed as $\mathrm{F1}(s+1,c)/K$. For case (a), this is the probability that both cases (b) and (c) do not happen, which is expressed as $(1-(P[S_{s,p,c}]+\mathrm{F2}(s,c)+\mathrm{F1}(s+1,c))/K)$. These probabilities are then weighted with the pass probabilities for each case as shown in Fig. 4. With regard to Fig. 2, an example transition from state 7 to state 8 has the probability of case (c) as $P[S^{1}_{1,2,4}]/K$, and the probability of case (b) as $\mathrm{F1}(2,4)/\mathrm{K} = (\mathrm{P}[\mathrm{S}_{3,1,4}] + \mathrm{P}[\mathrm{B}^{2}_{2,1,4}])/\mathrm{K}$. However, with a fault in a stage of the DN, Eq. 15 is derived using the probability that a request is not routed to the faulty SE being $(\frac{KN}{2} - 1)/(\frac{KN}{2})$.

$$P[H^b_{p,c+1}|S^b_{n-1,p,c}] = 1 \quad (s = n-1) \tag{18}$$

Eq. 18 means that the request always gets through the OM when it is in the last stage of a RN. From Fig. 2, state 9 is an example of such a state.

3.4.2 Request is currently in a Blocked state

$$\left.\begin{array}{rcl} P[S^b_{0,p+1,c+1}|B^b_{0,p,c}] &=& 1 - 0.5(\mathrm{F1}(1,c)) \\ P[B^0_{0,p_{\mathrm{new}},c+1}|B^b_{0,p,c}] &=& 1 - P[S^b_{0,p+1,c+1}|B^b_{0,p,c}] \end{array}\right\} \tag{19}$$

where $s = 0$, $p \neq K$.

$$P[L^b_{p,c+1}|B^b_{0,p,c}] = 1 \quad (s = 0, p = K) \tag{20}$$

Eq. 19 expresses the probability of a request being in a position to be re-routed via an alternate path (one of the outputs of a 2×4 SE in stage 0). This is the probability that a particular output of the SE in stage 0 is not occupied. Recall that $\mathrm{F1}(1, c)$ is the probability that, in cycle c, there is another request at some stage ahead of stage 0. In addition, there is 50% chance that this request has made use of the particular output required. An example is the transition from states 11 to 35. Eq. 20 applies when a request has failed on all paths, so it has to return to its IM; examples include states 15 and 19.

$$\left.\begin{array}{rcl} P[S^b_{s,p_{\mathrm{new}},c+1}|B^b_{s,p,c}] &=& 1 - 0.5(\mathrm{F1}(s+1,c)/(2^s)) \\ P[B^s_{s,p_{\mathrm{new}},c+1}|B^b_{s,p,c}] &=& 1 - P[S^b_{s,p_{\mathrm{new}},c+1}|B^b_{s,p,c}] \end{array}\right\} \tag{21}$$

where $1 \leq s \leq l-1$ and $p \bmod 2^{(l-s)} \neq 0$.

$$P[B^b_{s-1,p,c+1}|B^b_{s,p,c}] = 1 \quad (1 \leq s \leq l-1,\ p \bmod 2^{(l-s)} = 0) \tag{22}$$

Eq. 21 follows the same argument as Eq. 19 except that the probability that there is another request already occupying the required output is reduced by the factor of 2^s where s is the stage number. As an example, this refers to a transition from state 6 to state 7 in Fig. 2 in which the probability of another request already being ahead is the probability of its being in state 3 (*i.e.* a request in stage 2). Recall that there are 2^s planes in stage s $(l-1 \leq s \leq 0)$. As in Fig. 1, suppose that there is another request in stage 2, the 2×4 SE in stage 1 where this request has just successfully set up a path could have been either the one on the left plane or the other on the right. Thus, there is a 50% chance that this particular request has been occupying the same SE in which the considered request is going to be re-routed. Eq. 22 applies when a request has failed on all the paths that are available at a particular stage, thus it has to go back to a previous stage where it can try other paths. Examples include states 12 and 26.

$$P[B^b_{s-1,p,c+1}|B^b_{s,p,c}] = 1 \quad (l \leq s \leq n-2) \tag{23}$$

When a request is in a Blocked state at a stage in the RN, it simply returns to the previous stage.

3.4.3 Request is currently in a Hold state

(1) Request arrived at an OM without being blocked

$$P[H_{p,c+1}|H_{p,c}] = 1 \tag{24}$$

(2) Request arrived at an OM having been blocked

$$P[H^b_{p,c+1}|H^b_{p,c}] = 1 \tag{25}$$

3.4.4 Request is currently in a Lost state

$$P[L^b_{p,c+1}|L^b_{p,c}] = 1 \tag{26}$$

3.5 Derivation of P_{A} of a Proposed Switch

P_{A} can be computed by adding all the probabilities that the request is in a Hold state.

$$P_{\mathrm{A}} = \Big(P[H_{1,m}] + \sum_{p=2}^{K}\sum_{b=0}^{n-2} P[H^b_{p,m}]\Big)/P[S_0] \quad (c = m) \tag{27}$$

where m is the last cycle of the path-establish phase.

3.6 Simulation

Simulation was conducted in order to verify the analysis. The switch was simulated under the same assumptions as those adopted in the analytical model. Two independent runs were carried out, each consisting of five batches—altogether giving ten values to be averaged for each simulation point. Each batch consisted of 10,000 cycles. In each run, an additional 1,000 cycles were completed before data was collected in order to avoid transient affects. The 90% confidence intervals are very tight and therefore will not be shown. However, due to time limitations, only 16×16 and 32×32 switches were modelled.

4 Results and Discussion

Fig. 5 shows the P_{A} of a 16×16 with l=2 switch in re-routing mode fault-free and with a single fault present in various stages. Results for a 32×32 switch are depicted in Fig. 6. It is clear that the location of a fault determines the degradation of switch performance. It should be noticed that the performance of a 32×32 switch is less affected than a 16×16 switch. Fig. 7–9 compare the P_{A} of the switch with a single fault present as obtained from both analysis and simulation. They show reasonable agreement between the two. In both cases though, the analytical results gave a higher P_{A} than the simulation, which we presume to be due to failure of the independence assumptions. Curves of P_{A} of the proposed switch with a single fault in various stages as function of the number of stages are plotted in Fig. 10. Clearly, Fig. 10 indicates that the larger the switch size, the less affected the switch performance is.

5 Conclusions and Future Work

In this paper, we have presented an analysis of a fault tolerant ATM switch design based on the use of a number of routing networks in parallel when rerouting is performed and a single fault is present in various stages of the switch. The analysis is based on the state diagram method. The results obtained from the analysis were compared with the simulation results. It was shown that they were in reasonable agreement. These results also show that the degradation in performance of the switch depends on the location of a fault and that the performance is less affected with larger switches. There are a number of other aspects to be investigated in future. These include an analytical model for rerouting mode in the presence of multiple faults and reliability analysis of the switch.

Acknowledgements

The research described in this paper was undertaken by the first author (supervised by the second) as part of a Ph.D. project in the Dept. of Electronics Systems Engineering, University of Essex. The first author was initially supervised by Dr. E.A. Medova, and is sponsored by the Thai Government.

References

[1] Tobagi FA. Fast packet switch architectures for broadband integrated services networks. Proc. of the IEEE 1990; 78:133–167

[2] Venkatesan R, Mouftah HT. Performance analysis of multipath Banyan networks. Proc. IEEE International Conference on Communications 1992; 912–915

[3] Segkhoonthod S, Sinclair MC. Analysis of a fault-tolerant ATM switch based on a parallel architecture. Proc. UK Teletraffic Symposium, Old Windsor 1995; 22/1–22/9

[4] Segkhoonthod S, Sinclair MC. Re-routing analysis of a fault-tolerant ATM switch based on a parallel architecture. Third IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, Ilkely, 1995; 73/1–73/12

[5] Padmanabhan K, Lawrie DH. Performance analysis of redundant-path networks for multiprocessor systems. ACM Trans. Computer Systems 1985; 3:117–144

[6] Kumar PV, Reibman LA. Failure dependent performance analysis of a fault-tolerant multistage interconnection network. IEEE Trans. Computers 1989; C-38: 1703–1713

[7] Wu CL, Lee ML. Performance analysis of multistage interconnection network configurations and operations. IEEE Trans. on Computers 1992; C-41:18–27

[8] Patel JH. Performance of processor-memory interconnections for multiprocessors. IEEE Trans. on Computers 1981; C-30:771–780

[9] Kumar M, Jump JR. Performance of unbuffered shuffle-exchange networks. IEEE Trans. Computers 1986; C-35:573–577

[10] Hamming RW. The Art of Probability for Scientists and Engineers. Addison-Wesley, 1992

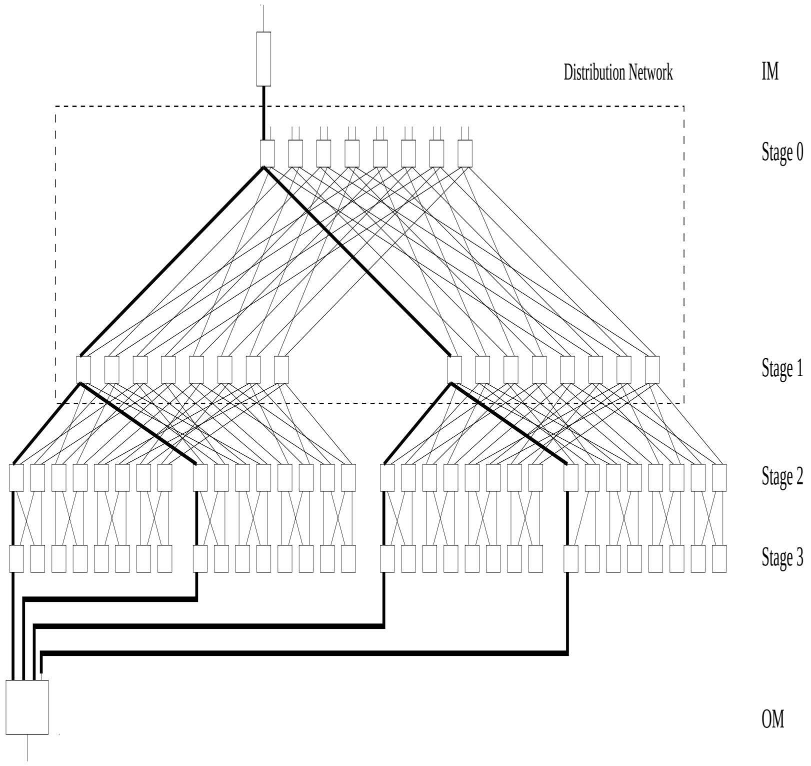

Figure 1: An $l = 2$, $K = 4$, 16×16 switch

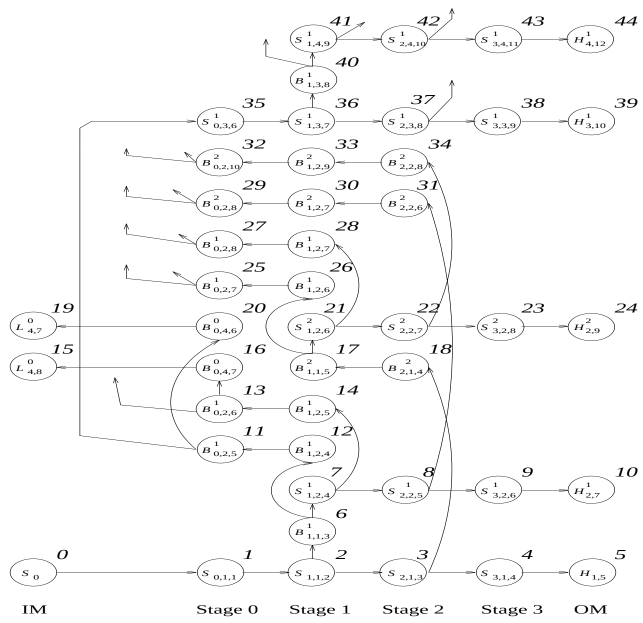

Figure 2: State diagram of an $l = 2$, $K = 4$, 16×16 switch with a fault in stage 2

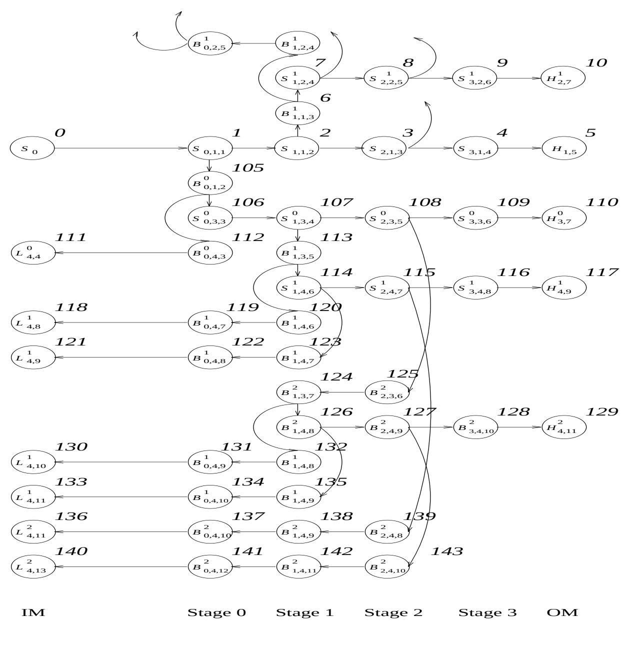

Figure 3: State diagram of an $l = 2$, $K = 4$, 16×16 switch with a fault in stage 1

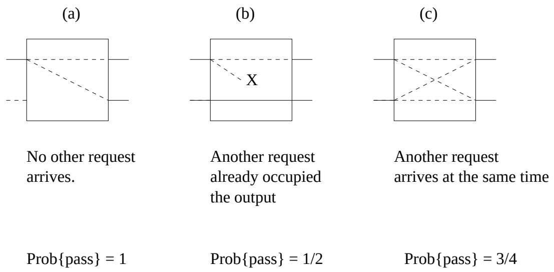

Figure 4: Probability of establishing a path in a 2×2 SE

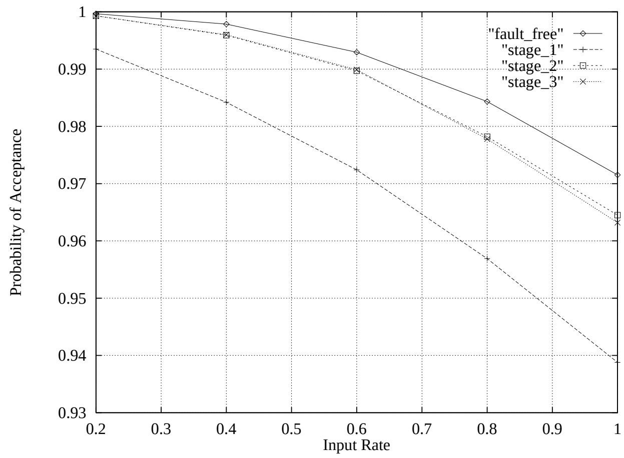

Figure 5: P_{A} of a 16×16 switch in re-routing mode with a fault in each stage (simulation results)

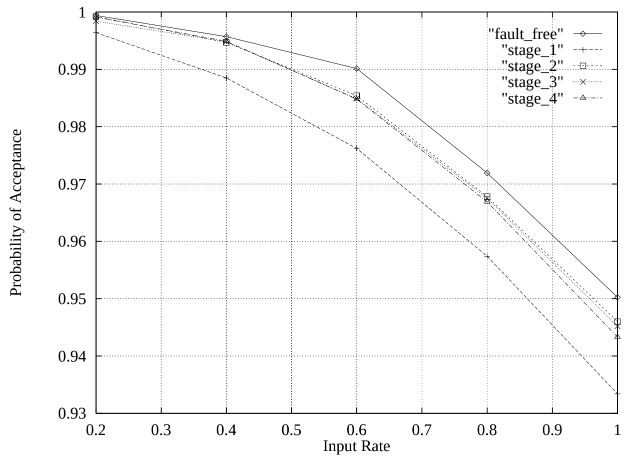

Figure 6: P_{A} of a 32×32 switch in re-routing mode with a fault in each stage (simulation results)

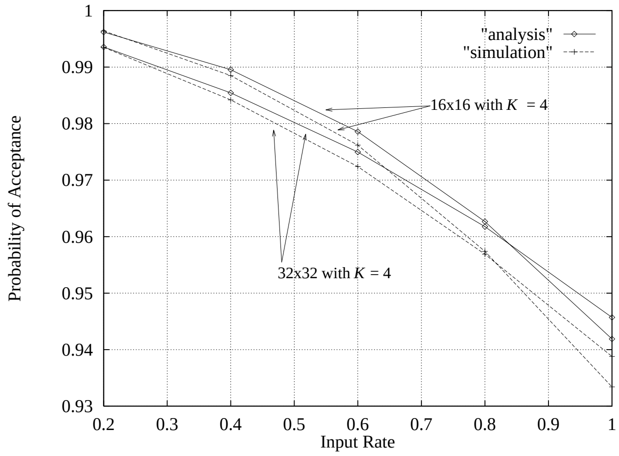

Figure 7: $P_{\rm A}$ in re-routing mode with a fault in stage 1 (simulation and analytical results)

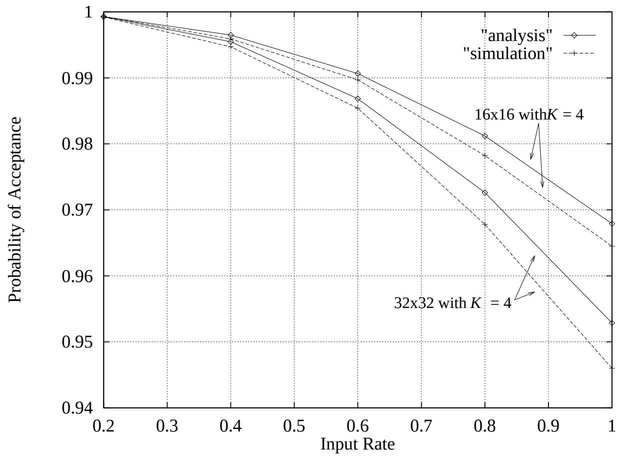

Figure 8: $P_{\rm A}$ in re-routing mode with a fault in stage 2 (simulation and analytical results)

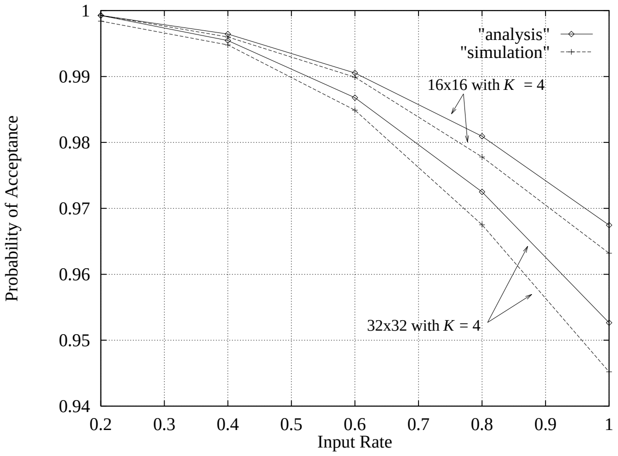

Figure 9: P_{A} in re-routing mode with a fault in stage 3 (simulation and analytical results)

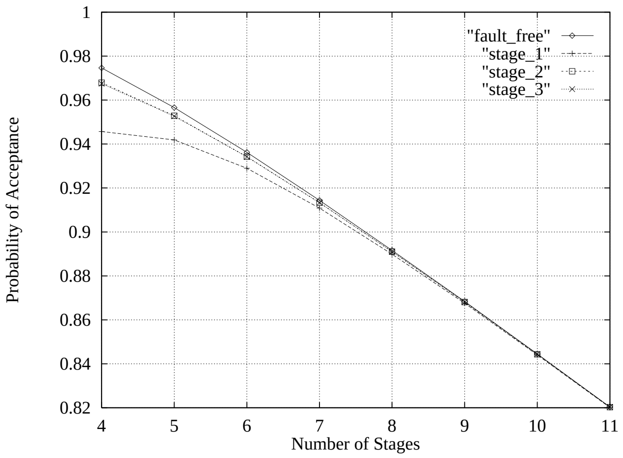

Figure 10: P_{A} of the proposed switch in re-routing with no faults and with a fault in each stage (analytical results)