01 Oct 2003

Modeling and Analysis of Fault Tolerant Multistage Interconnection Networks

Minsu Choi
Missouri University of Science and Technology, choim@mst.edu

Nohpill Park

Fabrizio Lombardi

Recommended Citation

M. Choi et al., "Modeling and Analysis of Fault Tolerant Multistage Interconnection Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 5, pp. 1509-1519, Institute of Electrical and Electronics Engineers (IEEE), Oct 2003.
The definitive version is available at https://doi.org/10.1109/TIM.2003.817906

Modeling and Analysis of Fault Tolerant Multistage Interconnection Networks

Minsu Choi, *Member, IEEE*, Nohpill Park, *Member, IEEE*, and Fabrizio Lombardi, *Member, IEEE*

***Abstract*—Performance and reliability are two of the most crucial issues in today's high-performance instrumentation and measurement systems. High speed and compact density multistage interconnection networks (MINs) are widely-used subsystems in different applications. New performance models are proposed to evaluate a novel fault tolerant MIN arrangement, thereby assuring performance and reliability with high confidence level. A concurrent fault detection and recovery scheme for MINs is considered by rerouting over redundant interconnection links under stringent real-time constraints for digital instrumentation as sensor networks. A switch architecture for concurrent testing and diagnosis is proposed. New performance models are developed and used to evaluate the compound effect of fault tolerant operation (inclusive of testing, diagnosis, and recovery) on the overall throughput and delay. Results are shown for single transient and permanent stuck-at faults on links and storage units in the switching elements. It is shown that performance degradation due to fault tolerance is graceful while performance degradation without fault recovery is unacceptable.**

***Index Terms*—Concurrent fault detection, diagnosis, instrumentation, multistage interconnection network (MIN), performance analysis, sensor networks.**

I. Introduction

COMMUNICATION between disparate unis is a stringent requirement in today's instrumentation and measurement systems. A variety of these applications can be found in fields such as energy physics computerized instrumentation [1], parallel workload characterization instrumentation [2], virtual instrumentation and measurement for distributed data acquisition, and parallel data processing [3]. One of the most emerging applications of multistage interconnection networks (MINs) includes distributed/parallel sensor network, in which MINs are commonly used as interconnection subsystems between processing elements and distributed sensor arrays [4]–[6].

Performance (measured in throughput and delay) and fault tolerance are two of the most crucial issues in the operation of these systems [7]–[9]. Also, concurrent testing is stringently required for complex digital instrumentation such as required for the control and maintenance of sensor networks. A widely used subsystem for implementing these systems is the MIN. MINs are switching subsystems for providing large connectivity as well as handling high volume data traffic [10]. Baseline and Banyan MINs are widely used for their generality and versatility [8].

Real-time fault detection and diagnosis is yet another requirement for high performance instrumentation and measurement systems; to provide and assure quality-of-service for real-time data processing and fusion is a necessity due to the large number of sensors and actuators involved in these systems. In this context, MINs have a primary role [8] and fault detection in either switching elements or data links is a prerequisite for their reliable operation. Fault diagnosis techniques for MINs can be found in [11]–[13]. In [13], it has been proved that single failure can be diagnosed and located in MINs with a number of tests which is dependent of the network size. In [14], it is proved that baseline interconnection networks in which the number of input and output lines of a switching element is 2, can be diagnosed with a constant number of tests (i.e., independent of network size). Fault diagnosis techniques such as those presented in [14] and [13] are based on the assumption that only a single switching element in the MIN is faulty. Further assumptions are that the MIN provides point-to-point connections, i.e. a path is established between two processing elements (one located at a primary input and the other located at a primary output), and that the MIN is not operational while testing is performed. This is generally applicable to off-line fault detection. A testing approach for off-line fault detection in MINs, which is applicable to packet switching, has been proposed in [12]. This approach is based on a single stuck-at fault assumption in which a fault can either prevent packet transmission, or affect the integrity of the data sent in the packet. The major disadvantages of this approach are that packet length is considerably increased compared with the case with no detection capability; moreover, the case of incorrect switching is not considered. [15] and [16] have dealt with approaches for multiple fault detection and location of MINs as applicable to manufacturing. An on-line fault detection technique identifies faults on-the-fly, i.e., while the MIN is operational without affecting normal operation. In [17], an approach has been proposed for testing MINs. This approach relies on the use of signature analyzers for on-line fault detection. This approach has general applicability and uses an unrestricted fault model. However, it does not fully accomplish concurrent fault detection within the MIN operation as a two-phase technique is implemented for diagnosing various types of single fault. This may result in a considerable fault latency. [7] also introduced fault tolerance in MINs by using a minimal number of additional stages.

To assure performance and reliability, their throughput and delay in the presence of faults in a MIN must be evaluated and

Manuscript received May 26, 2002; revised June 25, 2003.

M. Choi is with the Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409 USA (e-mail: choim@umr.edu).

N. Park is with the Department of Computer Science, Oklahoma State University, Stillwater, OK 74078-1053 USA (e-mail: npark@a.cs.okstate.edu).

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: lombardi@ece.neu.edu).

Digital Object Identifier 10.1109/TIM.2003.817906

verified at the design phase. Parametric simulation technique have been proposed for evaluating the performance of MINs [18]–[23]. However, the many MIN configurations restrict a broad application of parametric simulation. Occurrence of faults and introduction of fault tolerance in MINs are additional hurdles. Thus, analytical modeling has been advocated as technique to evaluate performance of MINs due to simplicity and scalability [18], [19]. In [18], throughput and delay of unbuffered, packet-switched, multipath, multistage networks have been estimated by approximation techniques such as probabilistic modeling and Monte Carlo simulation. [19] has evaluated throughput and delay of single-buffered and multi-buffered MINs by using queuing analysis. Exact models for the analysis of shared buffer ATM MINs with arbitrary traffic distribution has been reported in [24]. Reliability analysis of a non fault tolerant MINs has been also studied in [25] However, these works [18], [19], [24], and [25] have analyzed the performance of MINs without considering the occurrence of a fault as well as its detection and the recovery capabilities of fault tolerant MINs.

The objective of this paper is to propose new performance evaluation techniques for fault tolerant MINs whose operations involve concurrent fault detection and recovery. To derive simple yet realistic steady state equations for throughput and delay of fault tolerant MINs, novel queuing models are proposed. A fault tolerant MIN architecture with concurrent fault detection and recovery is also described and evaluated through the proposed models to further verify its effectiveness and accuracy. Last, an overhead analysis is provided to justify the benefit of the proposed fault tolerant MIN.

The organization of this paper is as follows: In Section II, we present a discussion of sensor fusion and the role of MINs in these applications. In Section III, a fault tolerant MIN architecture with concurrent testing and diagnosis is initially proposed. Then, general assumptions and definitions for performance evaluation of the proposed fault tolerant MIN architecture are discussed in Section IV. The performance evaluation of the proposed fault tolerant MIN in the presence of transient faults with and without concurrent recovery are given in Sections V and VI, respectively. Similarly, the performance evaluation of the proposed fault tolerant MINin the presence of permanent faults with and without concurrent recovery are shown in Sections VII and VIII, respectively. Analytical results and discussion are given in Section IX. Discussion and conclusions are iwithented n the final section.

II. MIN in Sensor Fusion Applications

In the past, MINs have been commonly utilized to provide connectivity between processors and memory units [20]. Recently, MIN have been extensively used for different application domains. Sensor fusion has recently received substantial attention due to its wide applicability to different systems which have grown in both scale and scope. Sensor fusion refers to the capability to control, acquire and manipulate data from different sensors as interface to the outside world. Fusion may occurs at different levels: data can be either globally or locally accumulated such that its analysis can be performed to extract salient features of a specific nature. Sensor fusion poses stringent requirements in terms of connectivity, real-time operation and modularity [6]. These requirements are related to the large number of sensors commonly involved in the data acquisition process. In most cases, the organization of these sensors changes over time and due to operational conditions; for example in a satellite, reconfiguration of sensors is of an absolute concern [4] due to the long mission time and the unmanned nature of equipment maintenance. Moreover, sensors may produce large amount of data to account for specific experiments, hence connectivity to computational resources must be provided well in excess of normal throughput rates. Fusion, moreover, is not static: the same sensors can be used in many environments to monitor different parameters [6]. Control of sensors through actuators adds a further degree of complexity as data must be shared by multiple processing entities.

Fusion also requires the reliable transmission of information to/from the sensors; control of the functions and monitoring of the sensing activities necessitate interconnection methods that must be easily embedded in hardware [26].

As extensively analyzed in the past [21], a MIN can be utilized to meet these requirements; modularity is accomplished on a stage basis such that a large number of sensors can be accommodated at modest overhead for scalability. A MIN also can be configured according to different switching configurations by simply programming the basic elements (switches) and rearranging the interconnects between them [4]. From a standpoint of performance, a MIN offers significant advantages: routing can be on a switched basis, so latency can be substantially reduced and local memory can be introduced in the switching elements to facilitate streaming of data over the interconnection network. This paper address a further feature of a MIN that deserves attention in sensor fusion applications, namely its tolerance capability in the presence of faults.

III. Fault Tolerant MIN Architecture

This paper considers multistage interconnection networks of the *Baseline* type [14]; this type of network connects N primary input (PI) lines to N primary output (PO) lines using $\log_2 N$ stages. Each stage is made of a number of switching elements (SEs). Each SE has two input and two output lines. Therefore, the number of switching elements per stage is given by $N/2$.

The proposed fault detection approach is based on a new format of the message transmitted by the SEs in the MINs. For the purpose of testability, we add extra subfields to packets. P is parity subfield which is used as general purpose parity check bits. C is checking subfield which consists of two bits. A is address subfield of $2 \times (\log_2 N - 1)$ bits. It is a position variant field, where the values of the field to process depend on the stage where the packet currently resides. D is data subfield of d data bits. Thus, a packet is composed of $P \bullet C \bullet A \bullet D$, where $\bullet$ stands for the concatenation operator (commonly used in describing packets in communication protocols [8]). The two bits in the C subfield achieve the same objective as in [9], namely to detect a stuck-at fault in a link and to denote the outcome of the testing process for stuck-at faults in the registers (storing the data bits of the packets).

The packet format described above requires the design of an SE capable of handling the subfields to accomplish concurrent fault detection. The architecture of the proposed SE is shown in Fig. 1. The switching element consists of four blocks, as follows.

- S *block*: Selection/acknowledgment block.
- D *block*: Comparison/detection block.
- C *block*: Testing data register/concatenation block.
- R *block*: Routing/switching/C subfield generation block.

The S block handles the incoming two packets from the input lines I_1 and I_2 (assuming 2×2 SEs) as well as any time-out condition. In Fig. 1, all registers used in the switching element are also shown; note again that all these registers operate as FIFO and are connected to the C block to provide the desired concatenation for the packets. The incoming packets are divided into appropriate subfields such that fault detection can occur. This task is accomplished by the D block. The D block compares the A and C subfields of the two packets on a bit-by-bit basis (note that hereafter the term *disagreement* does not refer to the output of an EX-OR gate, but to the difference between expected and actual values for the two bits either equally positioned in two string or adjacent). After the packets are processed in an SE, the first two bits of the A subfields of the two packets are removed. When a disagreement is recorded, a *FAIL* signal is issued from each block to S block to submit a negative acknowledgment to the preceding SE (consider Fig. 1).

The operation of the proposed switching element is given as follows: for two incoming packets P_1 and P_2 (on I_1 and I_2, respectively), the P subfield of each is checked first and then the bits in the C subfields are received next. These bits are compared; if no disagreement occurs, the C bits for the outgoing packet are subsequently generated by the C block using the outcome of the data register test. Bits are shifted-in and out of all registers (thus generating the C subfields for the outgoing packets), while at the same time the A subfields of the incoming packets are analyzed for the test of validity of preceding and current SEs.

Comparison on a bit-by-bit basis for the A subfields is also performed. If no fault is recorded by disagreement, the block R accomplishes the required switching between input and output lines in the switching element. The A subfields of the outgoing packets are thus generated by removing the first two bits of the A subfields of the incoming packets. The switching element is thus ready to acquire the D subfields at the C block. These subfields are concatenated to the previously generated C and A subfields and routed to the appropriate output lines according to the switching mode of the element.

For the sake of flexible and reliable rerouting, redundant links are added to the legacy MIN as shown in Fig. 2. Transient or permanent SE malfunctions can be circumvented by utilizing them.

IV. General Assumptions and Definitions

In the proposed model, transient faults can be recovered by resubmission and rerouting. When faults are present in the MIN without fault recovery scheme, packets which are diagnosed as faulty, are purged from the system. In the fault diagnosis process, a *FAIL* signal is issued from each comparison and testing block to issue a negative acknowledgment when a fault (disagreement) is detected.

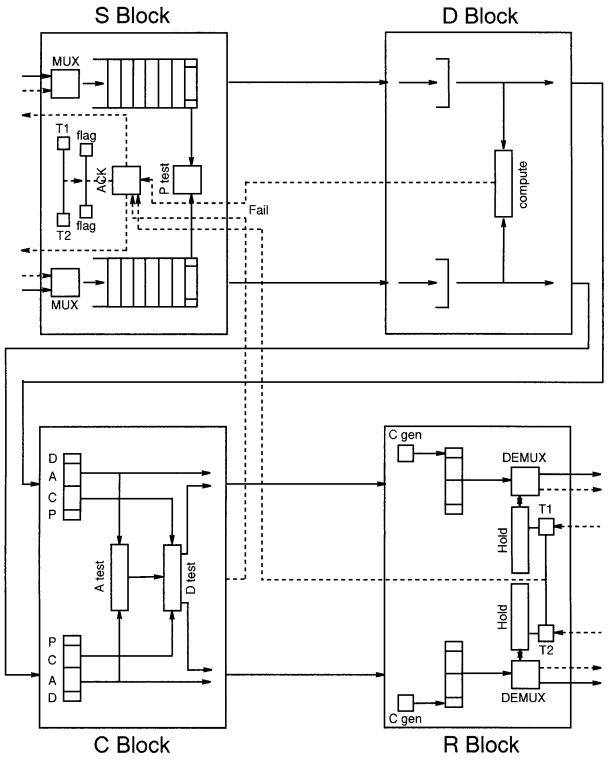

Fig. 1. Switching element of the proposed MIN.

The analysis is focused on the modeling of fault recovery operation so it is performed at the SE conducting a fault recovery operation. Virtually, only fault-free packets are included as inputs into each SE and faulty packets are excluded from input into succeeding SE at a constant fault rate at the SE conducting fault recovery operation.

A new parameter a to implicate the effect of faults into the model is introduced. A constant fault rate (P_f) for departing packets from each SE is assumed. The departure rate without a fault for a nonblocked packet is given by a which is derived by $1 - P_f$. This rate a for a nonblocked packet is equal to the probability of receiving a positive acknowledgment.

Under the assumption of a single fault per packet, only the affected packet is purged from the system. However, when faulty packets are found under a multiple fault assumption, the subsequent stages must purge the corresponding packet pair from the queue.

When a faulty packet occurs in the first stage of the network, this may result in a large number of subsequent packets being purged from the MIN. The closer the packet approaches the primary outputs prior to the occurrence of a fault, the smaller is the number of subsequent packets to be purged from the queues. Therefore, data loss decreases correspondingly as the packet moves to the latter stages.

Faulty packets are not included in the incoming queue for all succeeding stages since we assume that they are discarded or recovered at the driving SE at the rate of $1-a$ and a, respectively. If it succeeds in retransmission for a transient fault, it resumes

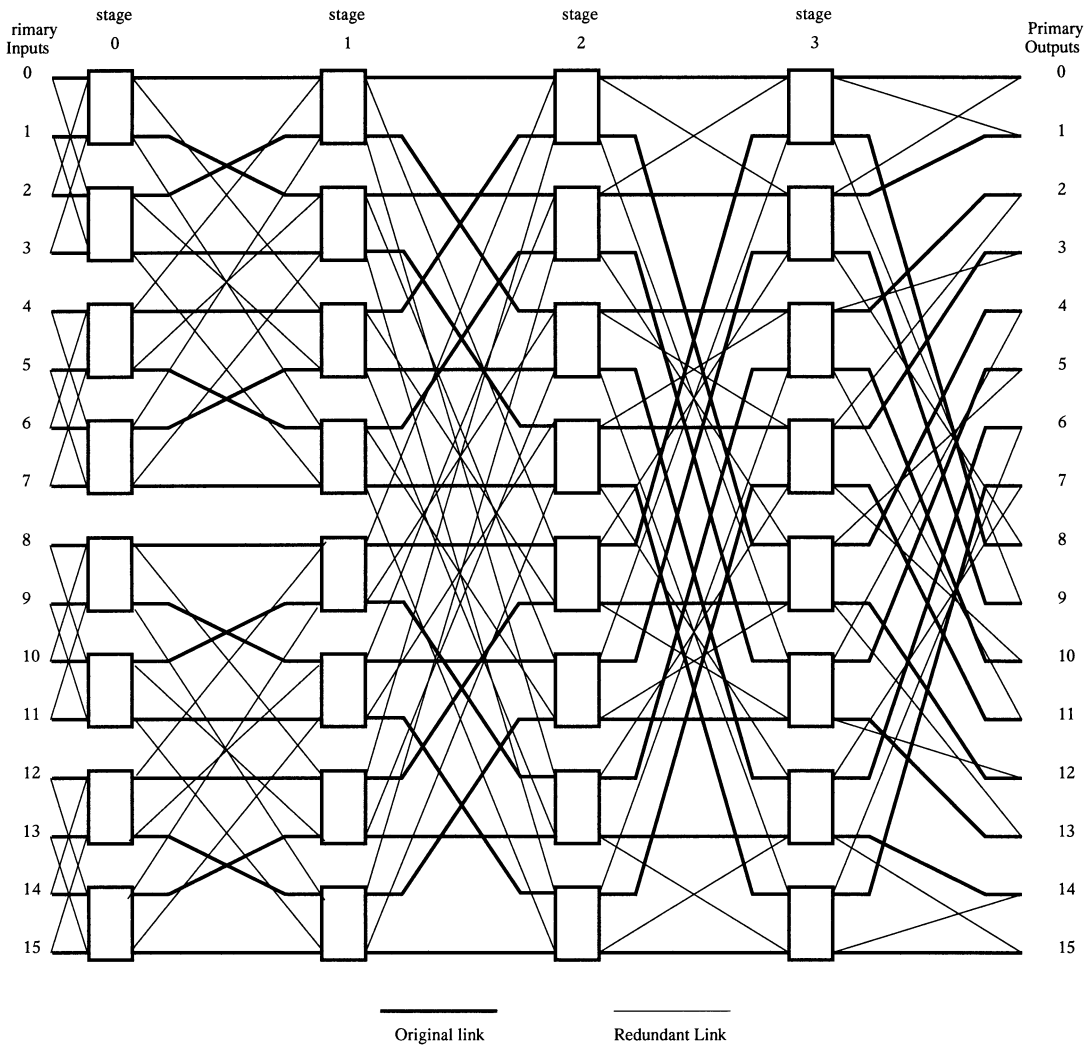

Fig. 2. Proposed fault tolerant MIN with redundant links (16 × 16).

its normal routing path. However, if it fails in retransmission, it is discarded in the nonfault-tolerant MIN or recovered through rerouting in the proposed fault-tolerant MIN. This creates two sets of input (throughput) for each queue in an SE.

- q_n: the input rate of normal packets at each queue of an SE.
- q_f: the input rate of rerouting packets that have been detected to be faulty in the fault-tolerant MIN. Notice that without recovery scheme, this rate of inputs will be discarded and not included into the input rate.
- T_n: the throughput of normal packets at each queue of an SE.
- T_f: the throughput of a queue of an SE along the redundant link in the fault-tolerant MIN. Notice that without recovery scheme, this rate of throughput will be discarded and not included into the throughput.
- Hence, total q and T are composed as $q = q_n + q_f$ and $T = T_n + T_f$.

Given an exponential SE fault rate with mean P_f (the number of faults per unit time), a fault will affect the MIN in one of the following two ways.

1) At least one packet will have corrupted data in the fields. These types of faults include faults such as stuck-at-0 and stuck-at-1 bits.
2) At least one packet will be routed inappropriately due to a faulty mode in the processing switch.

General network assumptions are summarized as follows.

1) Packets are generated asynchronously at the PIs and succeeding stages. However, a time bound Δ is imposed on the arrival time to guarantee the correctness of the incoming packets. If a packet does not arrive within Δ, it is assumed to be corrupt or not sent by the switch in the preceding stage. The timer generates a *FAIL* signal for the timed out packets.
2) The fault rate for each SE is assumed to be constant throughout the MIN. For each packet, the probability of a fault in it is constant and independent of the position of the packet. Hence, the fault rate is equally distributed on the interconnection network. Cluster faults are not considered. If the whole MIN system is fabricated on a single chip, then faults tend to show locality with respect to defects, so clustering may be considered consequentially. However, most of today's MINs are

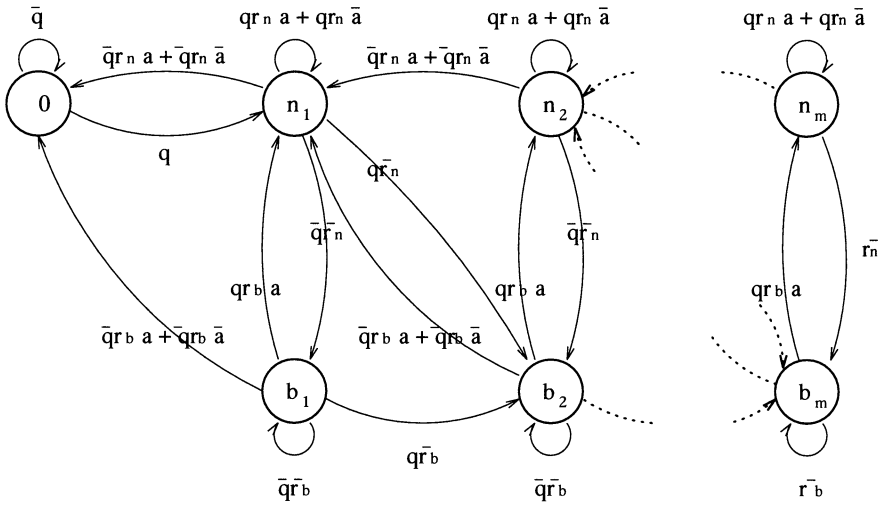

Fig. 3. State transition diagram of proposed TFWOR model.

made of independent components, such as switches and optical interconnects; therefore faults are assumed to be cluster-free in this paper.

3) To detect more than one bit error, the F field can be extended and existing error detection code (EDC) or error checking code (ECC) coding techniques can be employed to assure the packet data integrity. In practice, in the commercial-grade MINs, the average number of errors in a packet is well less than one bit. Hence, at most one fault per packet is assumed in this paper.

Based on the above assumptions, we perform the analysis at the individual SE level first and recursively analyze each connected SEs for the whole MIN. The states of each queue at an SE is defined as follows in the proposed model.

- State n_i: the state in which i number of packets are processed in a normal fashion without any blocking.
- State b_i: the state in which i number of packets are blocked in the queue by the *head of line* effect due to the blocking of the heading packet.
- State t_i: the state in which i number of packets are blocked in the queue by the *head of line* effect since the heading packet is acknowledged negatively and then trying to resubmit the packet through the redundant link.
- State s_i: the state in which i number of packets are blocked in the queue by the *head of line* effect since the heading packet trying resubmission of a packet through an redundant link is blocked from the new queue in the succeeding stage.
- State f_i: the state in which i number of packets reside in the queue in the status that the queue currently connected in the succeeding stage is detected to be permanently faulty. Hence, this state plays as a absorbing state in the presence of permanent faults.

V. Proposed Model With Transient Faults Without Recovery (TFWOR)

Basic single-buffer and multi-buffer MIN performance models are given in [19]. These models are simple and easily expandable for various versions of MIN operations, especially for the proposed fault tolerant MIN by adding new states and adjusting state transition probabilities. The state transition diagram for TFWOR is shown in Fig. 3 which is extended from the multi-buffer MIN performance model given in [19] and the state transition parameters are defined as follows.

- $q(i,t)(\overline{q(i,t)})$: $q(i,t)$ is the input rate at stage i at time t and $\overline{q(i,t)}$ is the complement of $q(i,t)$. In TFWOR, $q(i,t) = q_n(i,t) = 1 - q_f(i,t)$.
- $T(i,t)$: the throughput of an SE at stage i at time t. In TFWOR, $T(i,t) = T_n(i,t) = 1 - T_f(i,t)$.
- a: given a constant fault rate P_f of an SE, a is defined by $1 - P_f$.
- r_n, r_{nn}, r_{nb}, r_b, r_{bn}, r_{bb}, and P_0, P_n, P_b, and PN, PB: are defined the same way as in [19].

Note that $r_{nn}(i,t) = P_0(i,t) + 0.75SPn(i,t) + 0.5SPb(i,t)$ and $r_{nb}(i,t) = 0.25SPb(i,t)$ and $r_{bn}(i,t) = P_0(i,t) + 0.75SPn(i,t)$ and $r_{bb}(i,t) = 0.75SPb(i,t)$. Where, $SPn(i,t) = \sum_{j=1}^{m} PN(i,j,t)$ and $SPb(i,t) = \sum_{j=1}^{m} PB(i,j,t)$; i is the number of stage; j is the number of buffers in the buffer module; and t is the time. As in [19], r_n, r_b are calculated recursively from the last stage followed by calculations of P_{na}, P_{ba}, P_{bba} (as defined in [19]). Based on the proposed TFWOR model, the throughput and delay can be obtained as follows:

$$\text{Throughput} = SPn(n,t)r_n(n,t)a + SPb(n,t)r_b(n,t)a \quad (1)$$

$$D(i) = \lim_{t\to\infty} \frac{\sum_{j=1}^{m} j\,\{PN(i,j,t) + PB(i,j,t)\}}{T(i,t)} \quad (2)$$

VI. Proposed Model With Transient Faults With Concurrent Recovery (TFWR)

The states defined in TFWR are the same as TFWOR. We can effectively evaluate the state transitions by adjusting the state transition probabilities. The state transition diagram of TFWR is shown in Fig. 4 and the parameters are defined in the same way as in TFWOR except that $q(i,t) = q_n(i,t) + q_f(i,t)$ and

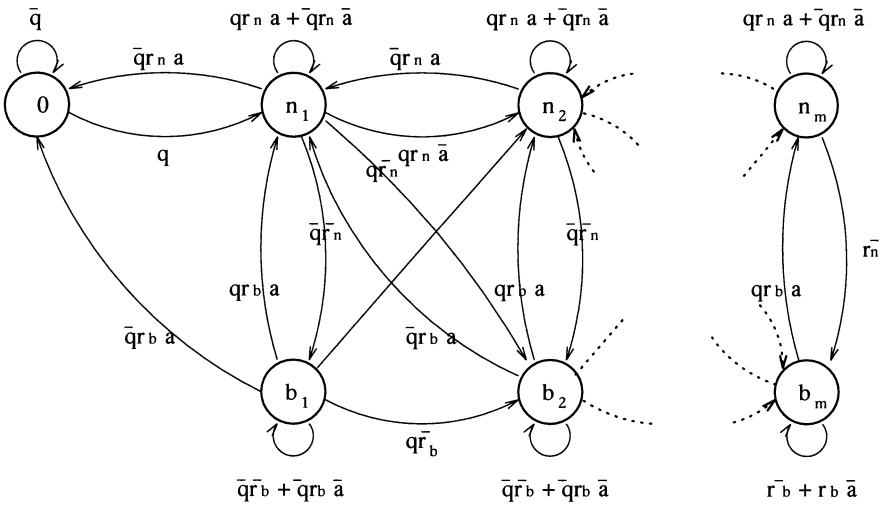

Fig. 4. State transition diagram of proposed TFWR model.

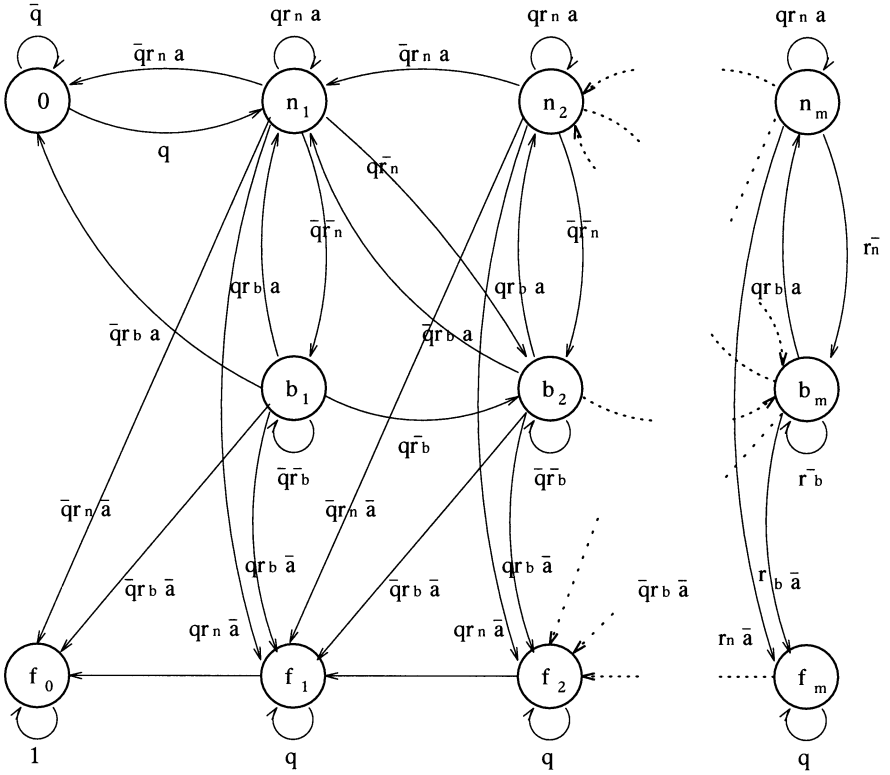

Fig. 5. State transition diagram of proposed PFWOR model.

$T(i,t) = T_n(i,t) + T_f(i,t)$ due to the fault recovery procedure through resubmission. Note that the variations are just the changes in the state transition probabilities due to the difference in network operations. Equations (1) and (2) can be used to calculate the throughput and delay of TFWR.

VII. Proposed Model With Permanent Faults Without Recovery (PFWOR)

In PFWOR model, we introduce absorbing states f_i as shown in Fig. 5. Once an SE is detected to be faulty, all the packets passing through it are discarded, since there is no way to recover the faulty packets. The state transition diagram of PFWOR is shown in Fig. 5 and $q(i,t)$, $T(i,t)$ are defined in the same way as in TFWOR since they both have no fault recovery scheme.

VIII. Proposed Model With Permanent Faults With Recovery (PFWR)

To evaluate the effect of the fault recovery on the performance, we need to introduce extra states t_i and s_i in PFWR model as shown in Fig. 6. In this model, the conflicts between the packets from original and redundant links should be considered. Each packet either from original or redundant link is assumed to have the same probability (i.e., $= 0.5$) to get through the conflict.

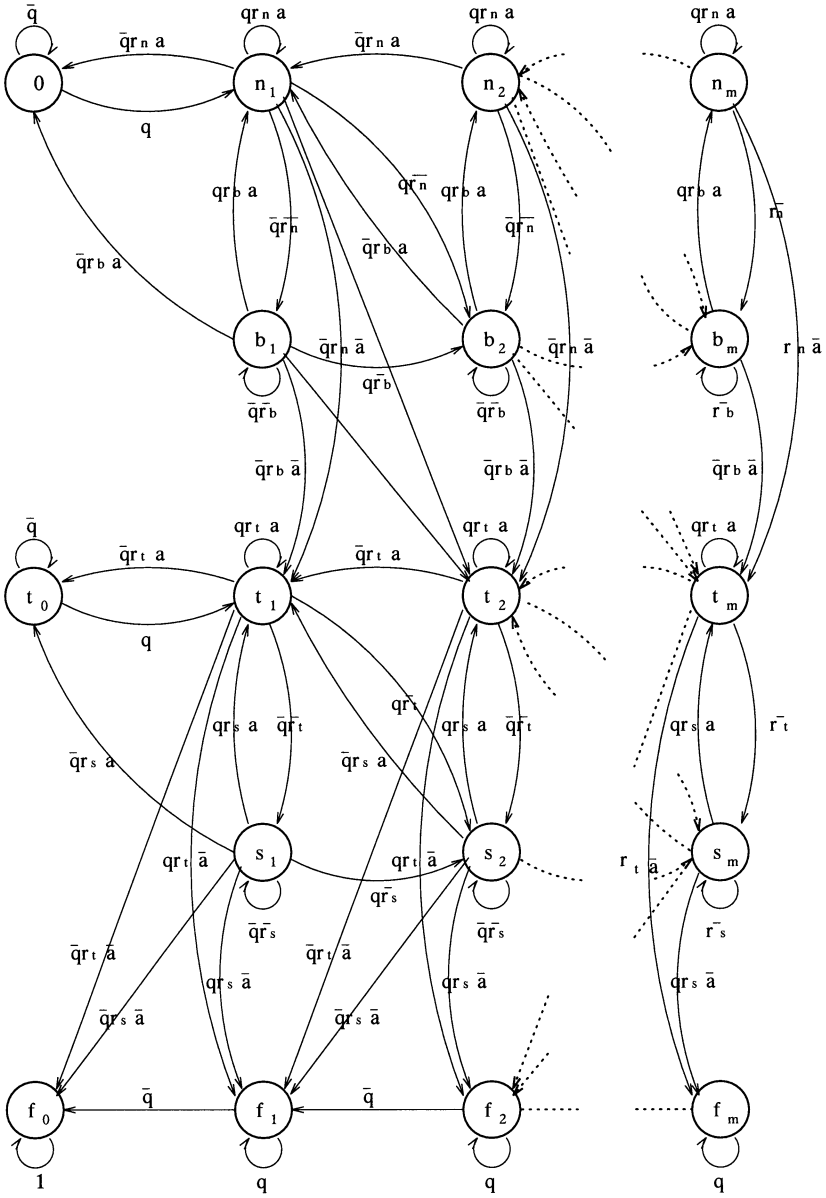

Fig. 6. State transition diagram of proposed PFWR model.

Also, note that $q(i,t)$ and $T(i,t)$ are defined as $q(i,t) = q_n(i,t) + q_f(i,t)$ and $T(i,t) = T_n(i,t) + T_f(i,t)$ since faulty packets are recovered.

Hence, PFWR is computed by the similar way as in TFWOR except that the computation of the probability to get through the conflicts between original link and redundant link is added to TFWOR.

IX. Analytical Results and Discussion

The performance of the MIN in terms of the normalized throughput and delay has been evaluated by using the proposed analytical models in the presence of transient and permanent faults. If the model reaches a steady state throughput, then we can derive a normalized delay by using the *Little's Law* as in transient faults case. The terms "*Normalized throughput*" and "*Normalized delay*" are defined in the same way in [19]. The results of the fault free MIN (*fftf* in the Figs. 7–12) are based on the model in [19]. A baseline network with 100 stages, 4 buffer modules, and 2×2 SE is considered.

Fig. 7–9 show the normalized throughputs of the MIN with transient faults by varying the value of the fault rate from 0.100 to 0.001. *tfwor* and *tfwr* are the plots of TFWOR and TFWR model, respectively. As expected, at high fault rate (0.1), there is a great loss in the throughput both in TFWOR and TFWR cases. However, as the fault rate decreases the loss in throughput is turning smaller.

In Fig. 10–12, the normalized delays of TFWOR and TFWR are shown. TFWOR shows that it takes much less delay at high fault rate due to the discarding of faulty packets which eliminates further conflicts. As the fault rate decreases, the delay of TFWR increases unlike the situation at high fault rate. TFWR

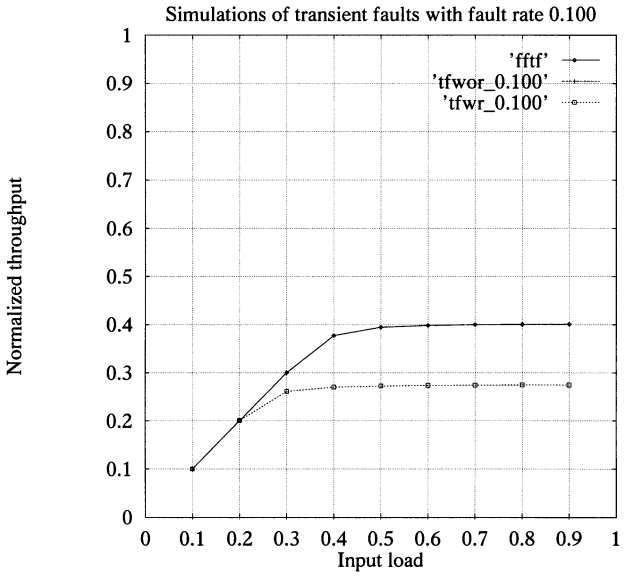

Fig. 7. TF with 0.100 fault rate (throughput).

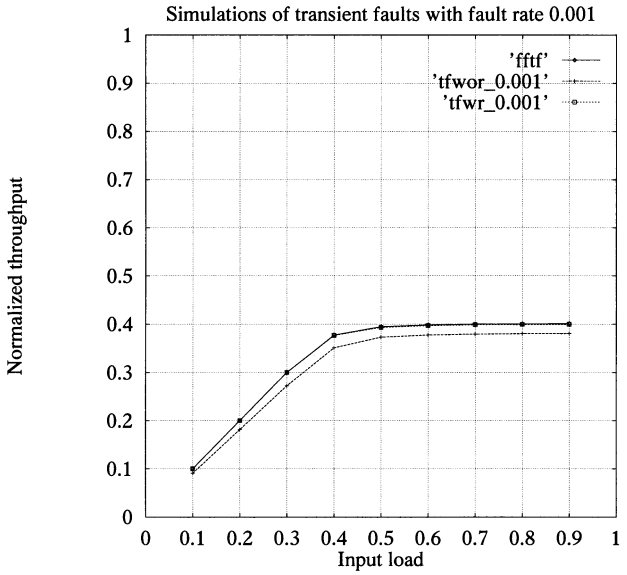

Fig. 9. TF with 0.001 fault rate (throughput).

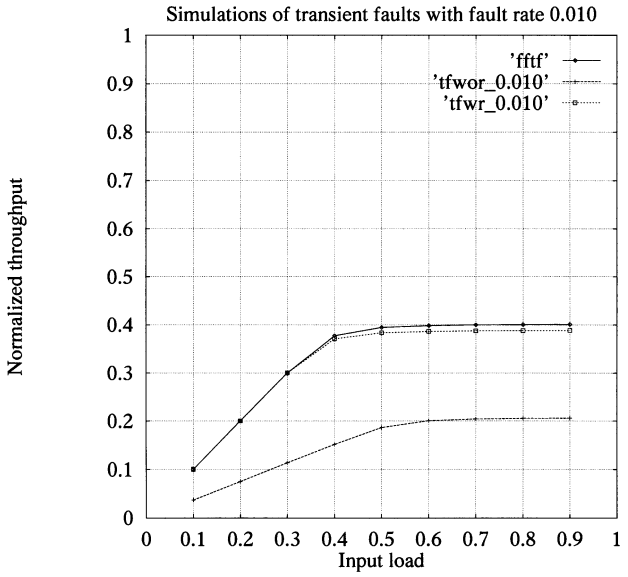

Fig. 8. TF with 0.010 fault rate (throughput).

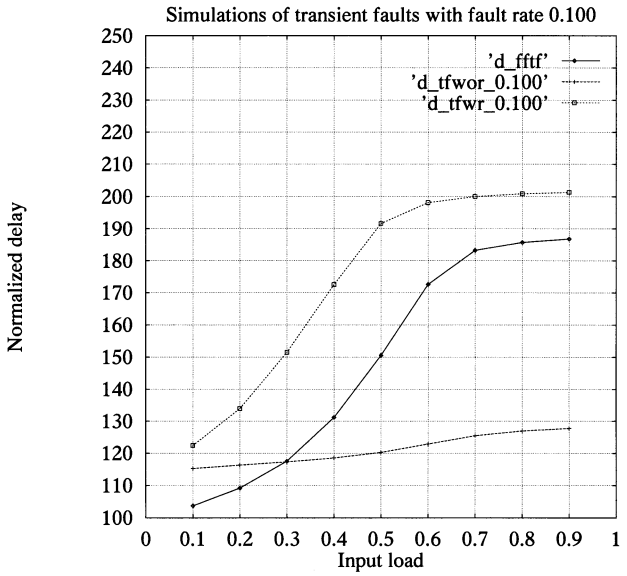

Fig. 10. TF with 0.100 fault rate (delay).

always takes longer delay than fault free MIN and TFWOR due to the delay caused by the retransmission of faulty packets. The delay of TFWR also decreases as the fault rate decreases.

Figs. 13–15 show the normalized throughputs of MIN with permanent faults. Due to the unsteady values of the normalized throughputs, we don't evaluate the normalized delays for the MIN with permanent faults. The normalized throughputs are time-dependent values and they are reaching the whole system failure in the long run. Hence, we look into the speed of the decrease of the throughput versus time, and compare those of PFWOR and PFWR cases.

At a high fault rate (i.e., $= 0.1$), both PFWOR and PFWR lose most of their throughputs; and as the fault rate decreases, they sustain some throughput which is reaching the whole system failure in the long run. At the fault rate of 0.01, PFWOR still loses most of its throughput. However, PFWR is achieving much higher throughput than that at 0.10 fault rate. At the fault rate of 0.001, PFWOR gets even much higher throughput in such a way that as the input load increases the peak throughput value is getting higher up to some almost saturated point. However, PFWOR still gets much less throughput and the speed of decreasing of the throughput is much higher than that of PFWR.

It can be observed that the analytical results of PFWOR and PFWR that at a certain fault rate (0.001), the proposed PFWR can enhance the performance greatly even with the extra conflicts caused by the recovery procedure and can further delay the whole system failure time greatly.

The overhead analysis on the proposed fault tolerant MIN is provided as follows.

1) *Hardware Overhead*: is mainly caused by the D block. This block consists of various multiplexers, few flip-flops (to generate the *FAIL* signal) and various comparators for bit-by-bit comparison of the appropriate subfields in the

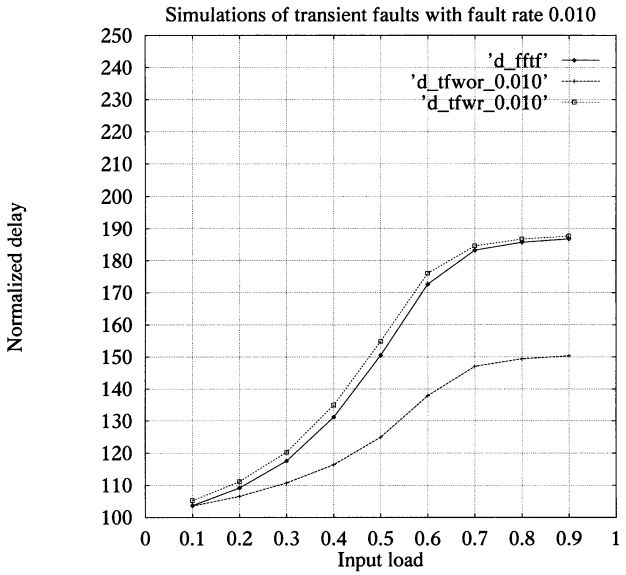

Fig. 11. TF with 0.010 fault rate (delay).

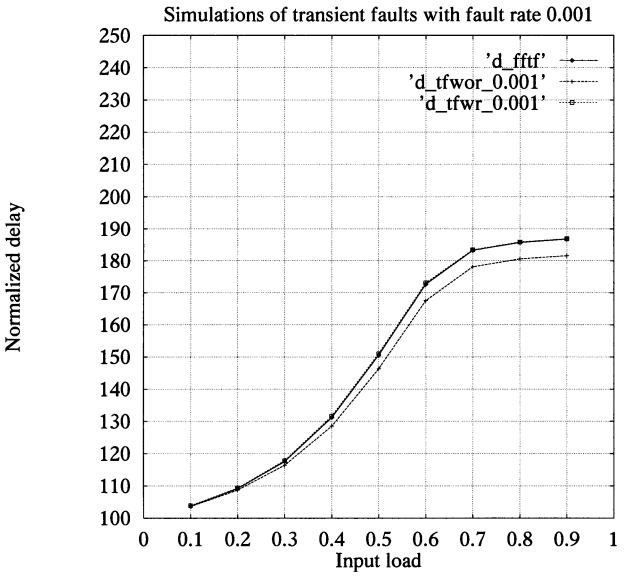

Fig. 12. TF with 0.001 fault rate (delay).

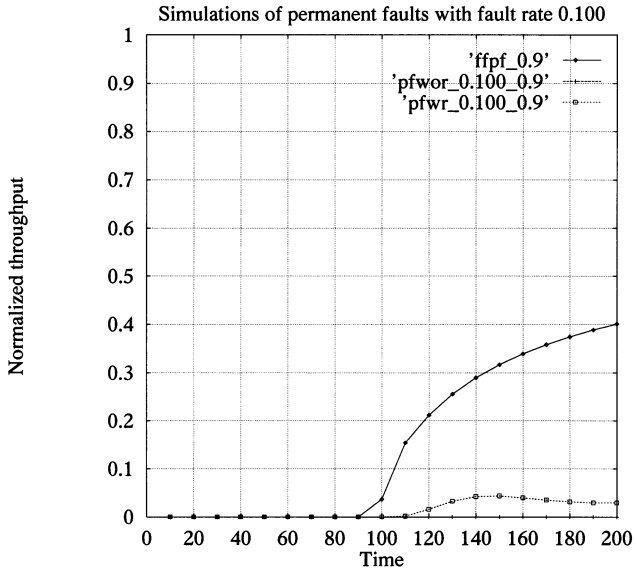

Fig. 13. PF with 0.100 fault rate and input load 0.9 (throughput).

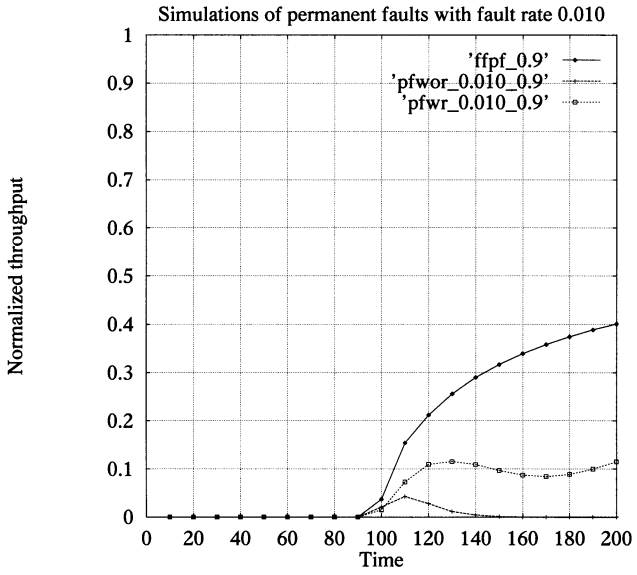

Fig. 14. PF with 0.010 fault rate and input load 0.9 (throughput).

incoming packets. If circuit complexity is calculated on a gate count basis, the D block accounts for only 6.2% (for $d = 8$) and 4.9% increase (for $d = 16$) approximately for $N = 64$ (where d is the number of bits in the D subfield in a packet). This is a reasonably acceptable overhead.

2) *Message Overhead*: is caused by the additional number of bits in a packet compared with a packet with no fault detection. In the new packet format, the added bits are only the 2 bits in the C subfield and 3 in the F subfield because the A and D subfields are also present in a packet with no fault detection capabilities. This is independent of N and d. Hence, the message overhead ratio is given by $O_{\text{packet}} = (2+3)/(d+2\times(\log_2 N))$. Table I shows the value of this overhead ratio (in percentile) for different values of d and N. The overhead ratio is reduced for larger values of d and N. Note that for the approach of [12] $O_{\text{packet}} = (2+3)/(d+2\times(\log_2 N))$, while for [17] it is dependent on the length of the signature.
3) *Transit Delay Overhead*: is directly proportional to two conditions, i.e. the number of bits and the required processing for concurrent fault detection. For the first condition, the delay overhead is equal to O_{packet} as presented in Table I. For the second condition, additional transit delay is accounted for the comparison operation in the D block as well as for testing the registers in C block. As the proposed SE operates in a pipeline mode for all bits of the incoming packets, comparison takes a single gate delay.
4) *Fault Latency*: is given by one transmission delay between stages as a fault is detected in at least one SE of the next stage. This differs from previous approaches [11] in which a fault is detected at the primary outputs of the MIN.
5) *Hard Core Component*: the proposed approach accomplishes concurrent fault detection without utilizing any hard

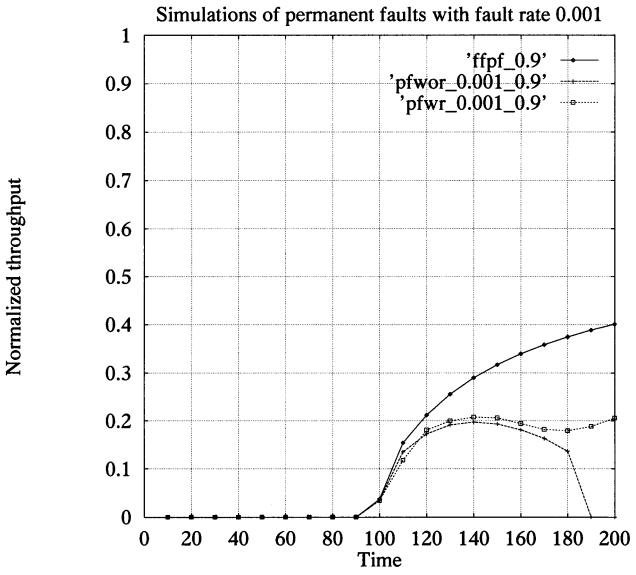

Fig. 15. PF with 0.001 fault rate and input load 0.9 (throughput).

TABLE I
OVERHEAD ANALYSIS RESULTS

N	d	$O_{packet}(Proposed)\%$	$O_{packet}([4])\%$
4	8	41.7	132.8
8	8	35.7	113.6
16	8	31.3	100
64	8	25.0	80.0
1024	8	17.8	56.8
4	16	25.0	160.0
8	16	22.7	145.6
16	16	20.8	132.8
64	16	17.8	113.6
1024	16	13.9	88.0

core element. Concurrent fault detection is accomplished in each SE following the single faulty element in the MIN. Therefore, it is not required to provide self-checking capabilities in each SE.

X. CONCLUSION

This paper has presented new performance models to evaluate the fault tolerant MIN as a subsystem in high speed and density parallel instrumentation and measurement systems. A switch architecture to realize the concurrent testing and diagnosis is shown and then the proposed performance models have evaluated the compound effect of the proposed fault tolerant operations such as testing, diagnosis, and recovery on the throughput and delay. A concurrent fault detection and recovery scheme for MINs for single stuck-at faults has been shown by using a new packet format, switching element architecture, and its communication protocol, to enable a generic approach to fault tolerance by resubmission and rerouting over the redundant interconnection links. The MIN achieves concurrent fault detection with fault secure operation at modest overhead in message length, fault latency and additional hardware. The fault recovery procedure based on the proposed concurrent fault detection approach is implemented on the MIN with low hardware overhead keeping the number of switching elements same. Results are shown for single stuck-at transient and permanent faults on links and storage units in switching elements. Both transient and permanent faults have been applied. It is shown that the performance degradation for the overhead due to concurrent testing, diagnosis and fault tolerance is quite graceful at low fault rate while the performance degradation without fault recovery or with fault recovery at high fault rate is unacceptable. The proposed work has established a sound foundation for designing high performance and reliable parallel instrumentation by using MINs with high confidence level, thereby ultimately realizing high quality-of-service in digital instrumentation such as real-time distributed/parallel sensor network, in which MINs are commonly used as interconnection subsystems between parallel processing elements and distributed sensor arrays.

REFERENCES

[1] G. Harangozo, "Two nondeterministic event building methods derived from barrel shifter," in *Annu. Simulation Symp.*, Apr. 1997, pp. 137–144.
[2] D. Marinov *et al.*, "Scowl: a tool for characterization of parallel workload and its use on splash-2 application suite," in *Int. Symp. Modeling, Analysis and Computer and Telecommunication Systems*, Aug. 2000, pp. 207–213.
[3] Z. Papp, H. J. Hoeve, and A. Bos, "A system architecture for distributed implementation of virtual measurement systems," *IEEE Eng. Comput.-Based Syst.*, pp. 18–24, Mar. 1999.
[4] M. Alderighi, F. Casini, S. D'Angelo, D. Salvi, and G. R. Sechi, "A fault-tolerance strategy for an FPGA-based multi-stage interconnection network in a multi-sensor system for space application," in *Proc. 2001 IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, Oct. 2001, pp. 191–199.
[5] K. Hirabayashi, T. Yamamoto, S. Hino, Y. Kohama, and K. Tateno, "Optical beam direction compensating system for board-to-board free space optical interconnection in high-capacity ATM switch," *IEEE J. Lightwave Technol.*, vol. 15, pp. 874–882, May 1997.
[6] K. Nishida, K. Toda, E. Takahashi, and Y. Yamaguchi, "A priority-based parallel architecture for sensor fusion," in *IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems*, Oct. 1994, pp. 105–112.
[7] C. Fan and J. Bruck, "Tolerating multiple faults in multistage interconnection networks with minimal extra stages," *IEEE Trans. Comput.*, vol. 49, pp. 998–1004, Sept. 2000.
[8] J. S. Turner, "Design of a broadcast packet switching network," in *Proc. IEEE Infocom*, 1986.
[9] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. IEEE/ACM 1st Annual Symp. on Comp. Arch.*, 1973, pp. 21–28.
[10] L. N. Bhuyan, B. Liu, and I. Ahmed, "Analysis of MIN based multiprocessors with private cache memories," in *Proc. ICCP*, 1989, pp. I-51–55.
[11] W. K. Fuchs, J. A. Abraham, and K. H. Huang, "Concurrent error detection in VLSI interconnection networks," in *Proc. IEEE FTCS*, 1983, pp. 309–315.
[12] W. Y.-P. Lim, "A test strategy for packet switching networks," in *Proc. Int. Conf. Par.*, 1982, pp. 96–98.
[13] C. Wu and T. Feng, "Fault diagnosis for a class of multistage interconnection networks," *IEEE Trans. on Comput.*, vol. C-30, pp. 743–758, Oct. 1981.
[14] F. Lombardi and W.-K. Huang, "On the constant diagnosability of baseline interconnection networks," *IEEE Trans. Comput.*, Dec. 1990.
[15] C. Feng, W.-K. Huang, and F. Lombardi, "Multiple fault detection and location in baseline interconnection networks," *IEEE Trans. Comput.*, vol. 41, pp. 1340–1344, Oct. 1992.
[16] K. M. Falavarjani and D. K. Pradhan, "Fault-diagnosis of parallel processor interconnection networks," in *Proc. IEEE FTCS*, 1981, pp. 209–212.
[17] J.-C. Liu and K. G. Shin, "Polynomial testing of packet switching networks," *IEEE Trans. Comput.*, vol. C-38, pp. 202–217, Feb. 1989.
[18] P. Sabalvarro, "Analytical modeling of multistage multipath networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 7, pp. 1059–1064, Oct. 1996.

[19] Y. Mun and H. Y. Youn, "Performance analysis of finite buffered multistage interconnection networks," *IEEE Trans. Comput.*, vol. 43, pp. 153–162, Feb. 1994.
[20] J. Kim, J. Park, H. Yoon, and J. W. Cho, "Fault-tolerant multicasting in MIN's for ATM switches," *IEEE Commun. Lett.*, vol. 2, pp. 331–333, May 1998.
[21] A. K. Somani and T. Zhang, "DIRSMIN: a fault-tolerant switch for B-ISDN applications using dilated reduced-stage," *IEEE Trans. Rel.*, vol. 47, pp. 19–29, Mar. 1998.
[22] Y. W. Leung, "On-line fault identification in multistage interconnection networks," *Parallel Comput.*, vol. 19, pp. 693–702, May 1993.
[23] T.-H. Lee and J.-J. Chou, "Testing the dynamic full access property of a class of multistage interconnection networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 5, pp. 1206–1210, May 1994.
[24] M. Saleh and M. Atiquzzaman, "Exact model for analysis of shared buffer ATM switches with arbitrary traffic distribution," *Proc. Inst. Elect. Eng. Comm.*, vol. 148, pp. 63–69, Apr. 2001.
[25] P. Mohapatra, C. Yu, and C. R. Das, "Allocation and mapping based reliability analysis of multistage interconnection networks," *IEEE Trans. Comput.*, vol. 45, pp. 600–606, May 1996.
[26] M. A. Figueriedo, P. H. Staken, T. P. Flatley, and T. H. Hines, "Extending NASA's processing to spacecraft," *IEEE Comput.*, vol. 32, pp. 115–118, May 1999.

Minsu Choi (M'02) received the B.S., M.S., and Ph.D. degrees in computer science from Oklahoma State University, Stillwater, in 1995, 1998, and 2002 respectively.

He is currently with Department of Electrical and Computer Engineering, University of Missouri, Rolla, as an Assistant Professor. His research mainly focuses on computer architecture and VLSI, embedded systems, fault tolerance, testing, quality assurance, reliability modeling and analysis, configurable computing, parallel and distributed systems, dependable instrumentation and measurement, and autonomic computing.

Dr. Choi is a member of Sigma Xi and Golden Key National Honor Society. He was a recipient of the Don and Sheley Fisher Scholarship, in 2000, the Korean Consulate Honor Scholarship in 2001, andt the Graduate Research Excellence Award in 2002.

Nohpill Park (M'99) received the B.S. degree and the M.S. degree in computer science from Seoul National University, Seoul, Korea, in 1987 and 1989, respectively, and the Ph.D. degree from the Department of Computer Science, Texas A&M University, College Station, in 1997.

He is currently an Assistant Professor in the Computer Science Department at Oklahoma State University, Stillwater. His research interests include computer architecture, defect and fault tolerant systems, testing and quality assurance of digital systems, parallel and distributed computer systems, multichip module systems, programmable digital systems, and reliable digital instrumentation.

Fabrizio Lombardi (M'82) received the B.Sc. degree (Hons.) in electronic engineering from the University of Essex, Essex, U.K., in 1977, the M.S. degree in microwaves and modern optics from the Microwave Research Unit, University College London, London, U.K., in 1978, as well as the Diploma in microwave engineering in 1978, and the Ph.D. degree from the University of London in 1982.

He is currently the Chairperson of the Department of Electrical and Computer Engineering and holder of the International Test Conference (ITC) Endowed Professorship at Northeastern University, Boston, MA. He was a faculty member at Texas Technical University, Lubbock, the University of Colorado, Boulder, and Texas A&M University, College Station. He received the Visiting Fellowship at the British Columbia Advanced System Institute, University of Victoria, Victoria, BC, Canada, in 1988, the TEES Research Fellowship from 1991 to 1992 and again from 1997 to 1998, and the Halliburton Professorship in 1995. He has been involved in organizing many international symposia, conferences, and workshops sponsored by organizations such as NATO and the IEEE, as well as Guest Editor in archival journals and magazines. His research interests are fault tolerant computing, testing and design of digital systems, configurable computing, defect tolerance, and CAD VLSI. He has extensively published in these area and has edited six books.

Dr. Lombardi received the International Research Award from the Ministry of Science and Education of Japan for the period from 1993 to 1999, the 1985/1986 Research Initiation Award from the IEEE/Engineering Foundation, a Silver Quill Award from Motorola in 1996, and a Distinguished Visitor of the IEEE Computer Society for the period from 1990 to 1993. Dr. Lombardi was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS from 1996 to 2000. Currently, he is the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTERS.