Traffic Analysis of f-cube Fault-Tolerant Routing Algorithm in Mesh Interconnection Networks

A. Nayebi[1], A. Shamaei[1], and S. G. Miremadi
[1]Sharif University of Technology
Azadi av., Tehran, Iran
{nayebi, a_shamaei}@ce.sharif.edu, miremadi@sharif.edu

Abstract

This paper presents a detailed traffic analysis of f-cube routing algorithm. Per-node traffic analysis illustrates the traffic hotspots caused by fault regions and provides a great assistance in developing fault tolerant routing algorithms. Moreover, the effect of a traffic hotspot on the traffic of neighbor nodes and global performance degradation is investigated, here. Currently, some coarse measures like global network latency are used to compare routing protocols. These measures do not provide enough insight of traffic distribution in presence of different fault regions. To analyze the per-node traffic, some per-node traffic measures are introduced here and one of them is selected for the rest of work. Different fault regions (single node, vertical line, horizontal line, and rectangular) in different places of a mesh network are simulated and the results are compared and interpreted. In an effort to gain deep understanding of the issue of traffic analysis of faulty networks, this paper is the first attempt to investigate per-node traffic around fault regions.

Keywords

Fault-tolerant routing, multi-computer networks, mesh topology, traffic distribution, f-cube routing.

1. Introduction

There exist several compute-intensive applications that require continued research and technology development to deliver computers with steadily increasing computing power [[1]]. The required levels of computing power can only be achieved with massively parallel computers, such as the Earth Simulator [[2]] and the Blue-Gene/L [[3]]. The long execution time of these applications requires keeping such systems running even in the presence of failures. However, the huge number of processors and associated devices significantly increases the probability of failure. In particular, failures in the interconnection network may isolate a large fraction of the machine, wasting many healthy processors that otherwise could have been used. Increasing clock frequencies leads to a higher power dissipation, which again could lead to premature failures [[4]].

It has been reported that NoC applications are prone to faults and have power dissipation restrictions [[5],[6],[7],[8]]. Mesh-connected topology due to its ideal simplicity and planarity is widely used in NoCs [[4],[6]]. Power dissipation is also an important issue in developing NoCs. Due to high level of complexity of VLSI circuit, uniform distribution of power dissipation and avoiding hotspots is so critical, too [[9]].

Fault regions [[10]] make some parts of interconnection network inaccessible and change the traffic distribution which potentially can make some bottlenecks and traffic hotspots in the network. These bottlenecks decrease the overall performance of the network and change power dissipation pattern. So, it's appreciated for any fault-tolerant algorithm to be associated with a detailed traffic analysis. One drawback is that achieving per-node terrific is a time consuming process and requires advanced tools. Therefore, many of published researches did not perform a detail analysis of traffic around fault regions [[10],[11],[12],[13],[14],[15],[16],[20]]. There is a brief analysis in [[21]].

In this paper we try to analyze traffic distribution around faulty regions in a most classical fault-tolerant routing algorithm known as f-cube [[10]] which is the basis of many other routing algorithms too. This aims to guide the reader through selection or rejection of this fault-tolerant routing algorithm for a specific application or improving this algorithm. This is achieved by providing information about detailed traffic distribution in case of presence of various faulty regions. Some measures for per-node traffic are provided and one of them is selected for the rest of the work. Simulations are performed for various fault region shapes, sizes, and positions. The results are interpreted for bottlenecks and effects of fault regions. Simulations are validated by investigation of batch results to ensure that system is in steady state. Interpretations show that results conform to the theory.

The rest of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 presents f-cube routing algorithm briefly. Section 4 introduces some per-node traffic measures which used in our analysis. Section 5 gives the simulation methods used and the results on the distribution of traffic around fault regions. Section 6 summarizes the work reported in this

```
Algorithm f-cube2 (Fault-Tolerant Routing around Block Faults in 2-D Meshes Procedure) :
1. Set and determine the message type (EW, WE, NS, or SN) based on the relative address of the destination.
2. At an intermediate node, a message is routed as follows:
    a. If the message has reached the destination, deliver the message.
    b. If the message has reached the destination column, set the message type to NS or SN.
    c. The message is forwarded along the dimension-order path if fault-free.
    d. The message has encountered a fault region. If this is the first hop in the fault ring, the message picks a direction to follow along the fault ring according to Fig. 2 and the relative location of the destination.
    e. If a dimension-order path is not free, the message continues in the same direction along a fault ring.
```

Fig. 1. Fault-tolerant routing around Block Faults in 2-D Meshes mesh

paper and the last section presents possible directions for future work.

2. Related work

Numerous deterministic fault-tolerant routing algorithms in meshes have been proposed in recent years [[10],[11],[12],[13],[14],[15],[16],[20],[22]] most of which augment the dimension-order routing algorithm to tolerate certain faults. Boppana and Chalasani proposed a fault-tolerant routing algorithm in mesh networks [[10]], or in mesh and torus networks [[11]]. The key idea of their algorithms is that, for each fault region, a fault ring or fault chain consisting of fault-free nodes and channels can be formed around it; if a message comes in contact with the fault region, the fault ring or chain is used to route the message around the fault region. This point implies that traffic around fault regions must be uneven and intensive. Deadlocks can be prevented by using four virtual channels per physical channel for deterministic (dimension-order) fault-tolerant routing. Their fault model is rectangle or special convex. In this work, two measures have been used to compare the overall performance of the proposed routing algorithm: "latency" and "bisection utilization". There is nothing in the presented work giving insight about traffic intensity around fault region. Sui and Wang[[15]], Zhou and Lau[[19]], Tsai [[17]], and Wu [[22]] proposed some fault-tolerant wormhole routing algorithms using various number of virtual channels.

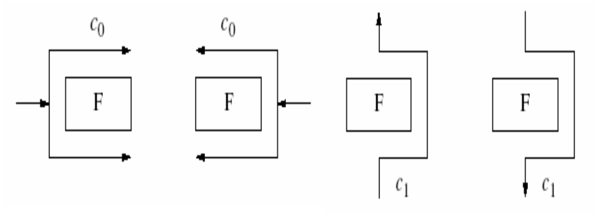


Fig. 2. Usage of virtual channels in f-rings (from [[24]])

Performance degradation in the above papers is mainly due to some bottlenecks in small areas of network especially at the corners of fault rings. We investigate this issue in details for f-cube in this paper and show how a bottleneck in a corner of fault region could propagate traffic to neighbor nodes and increase total network latency. This phenomenon is related to the nature of wormhole routing which has a great potential to propagate effects of regional bottlenecks into the entire network.

In [[18]] an appreciable analysis of traffic is performed and fairness of traffic is investigated but it is all about fault free networks.

3. The f-cube routing algorithm

"f-cube" routing is introduced in [[10]]. In this algorithm messages are assigned types based on the relative positions of the source and destinations and dimension-order routing. In a 2-D mesh, messages are typed as east-west (EW), west-east (WE), north-south (NS), or south-north (SN) based on the relative values of offsets in the first dimension. Routing is in dimension order until a message encounters a fault region. Depending on the type, the message is routed around the fault region (Fig. 2).

The direction around the fault region is selected based on the relative position of the destination node. The WE and EW messages use the c0 channels, with the NS and SN messages using c1 channels as shown. Since the topology is a mesh and there are no wraparound channels, there are no cyclic dependencies between c0 channels and also between c1 channels. The EW and WE messages may become NS and SN messages. However, the converse is not true. Thus, dependencies between channel classes are acyclic. A brief description of the routing algorithm is presented in Fig. 1

4. Per-node traffic measure

In order to deal with local traffic in a node, a measure must be declared which depicts how a single node may block the incoming messages and leads to a bottleneck. Some of the measures that can be considered are:

- AUC: Average Utilization of all incoming and outgoing physical Channels of a node.
- ANB: Average Number of Blocked messages in routing element
- AWB: Average Waiting time in Blocking queue
- AVC: Average number of used Virtual Channels in the node switch

In this paper we have used the AVC measure. In fact, this measure points the level of node switch utilization. The ANB measure is applied in only a single figure for comparison with AVC. Fig. 3 shows the effect of global traffic on different measures. These values are normalized to be comparable. As depicted in this figure, AUC measure is almost linear in unsaturated range. So, AUC indicates the traffic linearly and is not sensitive to reaching to saturation point. However, AUC depicts the traffic intensity as is. As illustrated in Fig. 3, ANB measure has the sharpest curve. It's almost zero in unsaturated region and rise up rapidly near the saturation point. This measure provides no insight on traffic propagation of a hotspot because of ignoring the traffic levels lower than a threshold. However, ANB diagnose hotspots more clearly. AWB is much similar to AUC. AVC has the benefit of highlighting the hotspots (reaching to saturation traffic level) with its exponential-like curve and also depicts the low level values of traffic (unlike ANB).

5. Simulation results

Simulations are performed using XMulator [[23]]. XMulator is a continues-time event-based flit-level simulator which is developed by the authors to achieve various detailed results. Virtual channels are time-multiplexed on a physical channel and like [[10]] only virtual channels that have messages to transmit use the physical channel in round-robin manner. Delays for switching and routing are ignored and only delay of physical channels is considered. Eight virtual channels are used for each physical channel. Injection channels and ejection channels have 8 virtual channels too. Message consuming and generation is not bandwidth limited. Message generation rate for each node is 0.0002 which keeps the network in normal functional state in all the scenarios and does not make the network saturated. Physical channel delay is 1 cycle and message length is typically chosen as 32 flits.

A 32 × 32 mesh is simulated, since radix 32 has been used in many previous studies. Uniform traffic pattern is generated by each node and message generation interval has exponential distribution which leads to Poisson distribution of number of generated messages per a specified interval. As mentioned before, AVC (average number of used virtual channels in each node) is considered as traffic measure. This is the time-averaged value of number of connected virtual channels in a switch element.

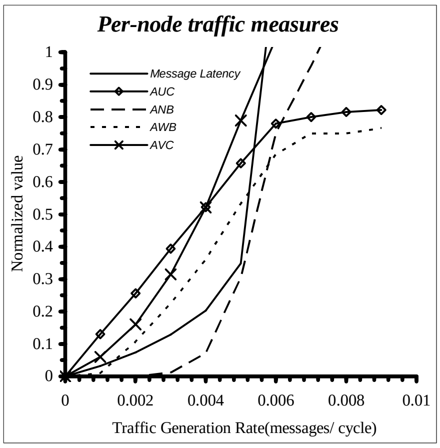


Fig. 3. Effect of global traffic on different per-node traffic measures.

Virtual channels are partitioned into two equal sets each containing 4 virtual channels which are belongs to c0 and c1 classes of f-cube algorithm.

5.1. Fault regions

Five fault regions are simulated: a 1 × 1 region, a 1 × 8 region, a 2 × 2 region, an 8 × 1 region, and an 8 × 8 region. These regions are chosen typically due to 32 × 32 size of network. All of these fault regions are placed in centre of network. Moreover, an 8 × 8 region is placed in east, west, north-east and south-east of network to investigate the effect of fault region position on the traffic distribution. These last fault regions which placed near network edges are not stuck to network edge and there is a line of nodes to the edges. This space is considered for better comparison of results.

5.2. Results

Results of simulation are depicted in
Fig. 4andFig. 5.

Fig. 4.a,b depict the fault-free results. As expected, the traffic in the middle parts of network is higher than corners. In circular regions from centre to corner the value of AVC is decreased continuously. This is due to irregularity of mesh topology and applied dimension order routing in fault free network.

Fig. 4.c depicts the presence of one faulty node at the centre of network. As illustrated in the figure, the semi-flat surface of
Fig. 4.a is substantially changed here and two peaks at south-east and north-east of faulty node are appeared. Moreover, two horizontal lines rise in rows 7 and 9.

Fig. 4.d depicts a 4 × 4 fault region.

Fig. 4.e depicts the effect of a 1 × 8 horizontal fault region.

Fig. 4.f depicts a 8 × 1 vertical fault region. Fig. 5.a depicts a centered 8 × 8 fault region. Fig. 5.b is same as Fig. 5.a, but ANB (Average Number of Blocked messages in routing element) measure is used. Fig. 5.c,d,e,f depict an 8 × 8 region cited at the east, west, north-east, and south-east regions respectively.

5.3. Interpretations

First, we explain the cause of appearing two peaks in the south-east and the north-east of any fault regions in the figures. As mentioned before, f-cube transmits NS (north to south) and SN (south to north) messages, clockwise and counter-clockwise respectively. This leads to un-even propagation of traffic and load the east part of the fault region more than the left part. These two peaks at the east corners show that this unfairness how deeply changes the traffic distribution. Second, the reason that corners have more traffic than edges is overlapping traffics of both edges on the corner. If an algorithm lets messages leave f-ring from middle part of the edges (with considering deadlock issues), bottlenecks must be eliminated considerably. Third, there are two horizontal lines which rise just above and below the fault region. This is because of nature of dimension order routing algorithm which first routes in X axis then in Y axis. When an east to west or a west to east message arrives at a fault region, it must go up or down and turn around the region but won't back to its previous row unless it has been reached to destination column. For example, if there is a block fault region from (13, 13) to (20, 20) the traffic intensity in rows 13 to 20 is decreased and traffic intensity in rows 12 and 21 is increased apparently (Fig. 5.a). Fourth, as depicted in the figures, centered fault regions make stronger traffic hotspots than fault regions cited in corners of the network. Moreover, comparing the Fig. 5.e with Fig. 5.f shows that f-cube performs better in case of vertical fault regions. Comparing the overall latency values confirm this result too. Fifth, comparing Fig. 5.a with Fig. 5.b shows that ANB measure presents the traffic hotspots more clearly but AVC measure presents the effect of a traffic hotspot on the neighbor regions.

6. Conclusions

In this paper, we analyzed the traffic distribution around faulty regions in meshed interconnection network. f-cube is chosen as a classical fault-tolerant routing algorithm. Four measures for per-node traffic proposed and AVC (average number of used virtual channels in switch) is selected for the rest of the work. Simulations are performed on a typical 32 × 32 meshed network with normal message generation rate. Results show that f-cube make a traffic hotspot at north-east and south-east of the fault ring. In a simulated 8 × 8 fault region cited in the centre of the network, traffic at the north-east corner was increased by a factor of 7.7 in comparison to fault-free network. This result is so considerable in designing power-aware NoCs. Moreover, it's apparently demonstrated that f-cube makes two horizontal lines with intensive traffic on top and bottom of fault region.

Our approach is the first attempt to provide insight on the traffic distribution around fault regions. Using the detailed information about the traffic hotspots, one can improve fault-tolerant routing algorithms.

References

[1] C.T. Ho, L. Stockmeyer, A new approach to fault-tolerant wormhole routing for mesh-connected par-allel computers. *IEEE Trans. on Computers, 53*(4), 2004, 427–438.

[2] Earth Simulator Center, http://www.es.jamstec.go.jp/esc/eng/index.html.

[3] IBM BG/L Team, “An Overview of BlueGene/L Supercomputer, *ACM Supercomputing Conference* , 2002.

[4] E. Nilsson, J. Oberg, Reducing Power and Latency in 2-D Mesh NoCs using Globally Pseudochronous Locally Synchronous Clocking. *lSSS'04, Slockholm, Sweden* , 2004, 176-181

[5] T. Dumitras¸S. Kerner, R. Marculescu, Towards On-Chip Fault-Tolerant Communication. Proc. *Asia & South Pacific Design Automation Conf.* (ASP-DAC), January, 2003.

[6] M. Pirretti, G.M. Link, R.R. Brooks, N. Vijaykrishnan, Kandemir, M. Irwin, Fault Tolerant Algorithms for Network-On-Chip Interconnect. *IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design* (ISVLSI'04), 2004, 46-52

[7] T. Simunic, K. Mihic,G. De Micheli, Reliability and Power Management of Integrated Systems. *Eu-romicro Symposium on Digital System Design* (DSD'04) , 2004, 5-11

[8] D. Wiklund, Development and Performance Evaluation of Networks on Chip. *Ph.D. thesis, Linköping University*, Sweden, 2005.

[9] D. Bertozzi, L. Benini, G.D. Micheli, Error Control Schemes for On-Chip Communication Links: The Energy–Reliability Tradeoff. *IEEE Trans. On Computer-Aided Design Of Integrated Circuits & Sys-tems, 24,* NO. 6, 2005, 818-831

[10] R.V. Boppana, S. Chalasani, Fault-tolerant wormhole routing algorithms for mesh networks. *IEEE Trans. Computers, 44*(7) , 1995, 848–864

[11] S. Chalasani,R.V. Boppana, Communication in multicomputers with non-convex faults. *IEEE Trans. Computers, 46*(5) , 1997, 616–622

[12] M.E. Gomez, J. Duato, J. Flich,P. Lopez,A. Robles, N.A. Nordbotten,T. Skeie, O. Lysne, A New Adaptive Fault-Tolerant Routing Methodology for Direct Networks. *HiPC 2004, LNCS 3296*, 2004, 462–473

[13] J. Wu, A simple fault-tolerant adaptive and minimal routing approach in 3-d meshes. *Journal of Computer Science and Technology. 18*(1) , 2003, 1-13

[14] J. Wu, Fault-tolerant adaptive and minimal routing in mesh-connected multi computers using extended safety levels. *IEEE Trans. on Parallel and Distributed Systems. 11,* (2) , 2000, 149-159

[15] J. Wu, D. Wang, Fault-tolerant and deadlock-free routing in 2-D meshes using rectilinear-monotone polygonal fault blocks. *The International Journal of Parallel, Emergent and Distributed Systems, 00(*0) , 2005, 1–14

[16] P.H. Sui, S.D. Wang, An improved algorithm for fault-tolerant wormhole routing in meshes. *IEEE Trans. Computers, 46*(9) , 1997, 1040–1042

[17] M.J Tsai, Fault-tolerant routing in wormhole meshes. *Journal of Interconnection Networks, 4*(4) ,2003, 463–495

[18] C. Izu, Throughput fairness in k-ary n-cube networks. *Proc. of ACSC2006,* ,2006.

[19] J.P. Zhou, F.C.M. Lau, Fault-tolerant Wormhole Routing in 2D Meshes. *Proc. International Symposium on Parallel Architectures, Algorithms and Networks,* Dallas/Richardson, Texas, USA. , 2000, 457-471

[20] J.P. Zhou, F.C.M. Lau, Fault-tolerant Wormhole Routing Algorithm in 2D meshes without Virtual Channels. *Proc. Of ISPA’2004* , 2004, 688-697

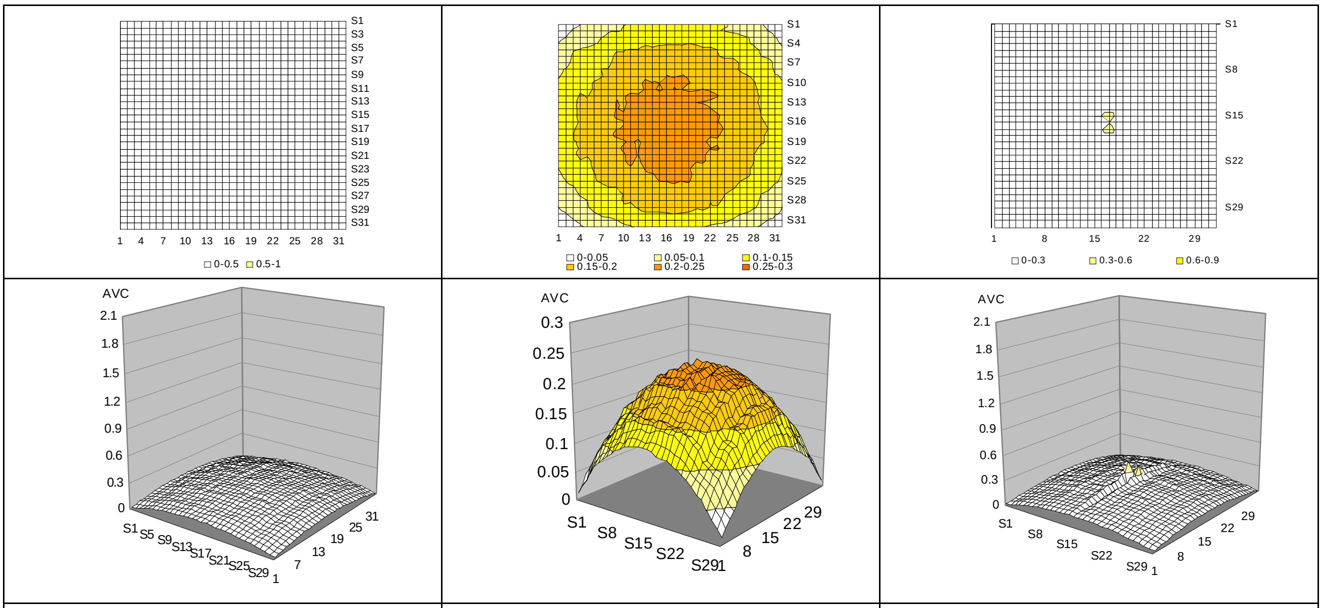
[21] V. Varavithya, J. Upadhyay, P. Mohapatra, An Efficient Fault-Tolerant Routing Scheme for Two-Dimensional Meshes. *International Conference on High Performance Computing,* 1995, 773-778

[22] J. Wu, A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model. *IEEE Trans. On computers, 52,*(9) 2003, 1154-1169

[23] A. Nayebi, H. Sarbazi Azad, A. Shamaei, S. Meraji, XMulator: An Object Oriented XML-Based Simulator. http://www.XMulator.org, 2006

[24] J. Duato, S. Yalamanchili, V. Ni, *Interconnection Networks, an Engineering Approach.* Morgan Kaufmann Publishers, USA, 2003

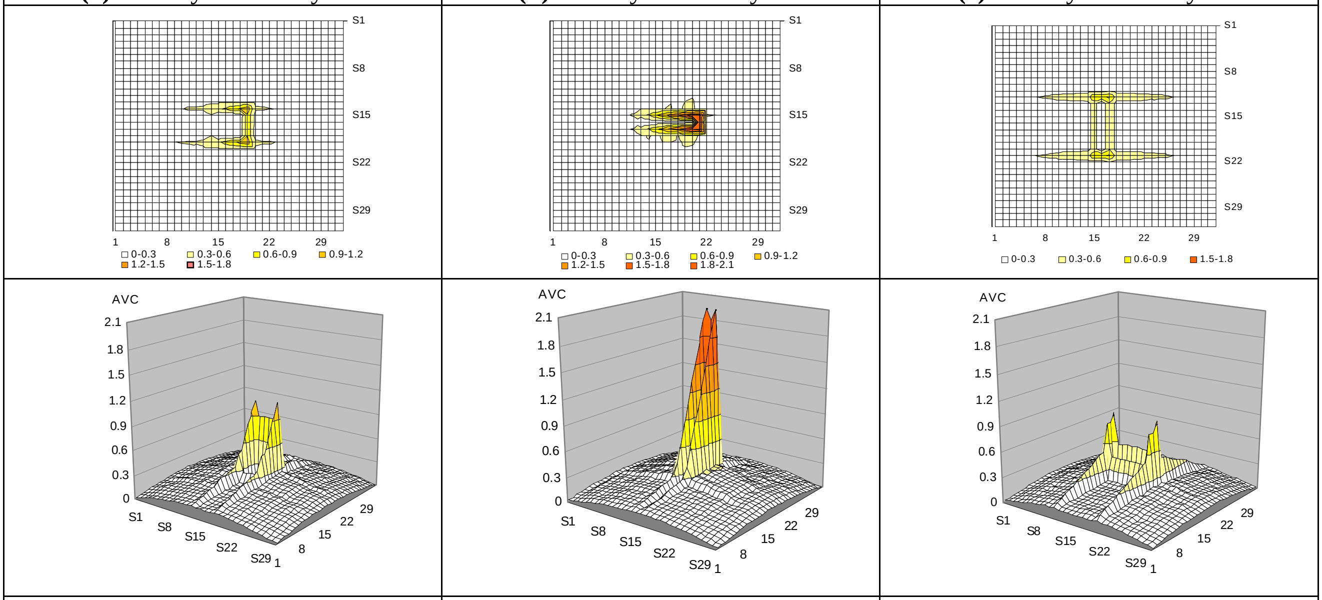
Appendix



(a) Latency = 56.51 cycles

(b) Latency = 56.51 cycles

(c) Latency = 56.85 cycles

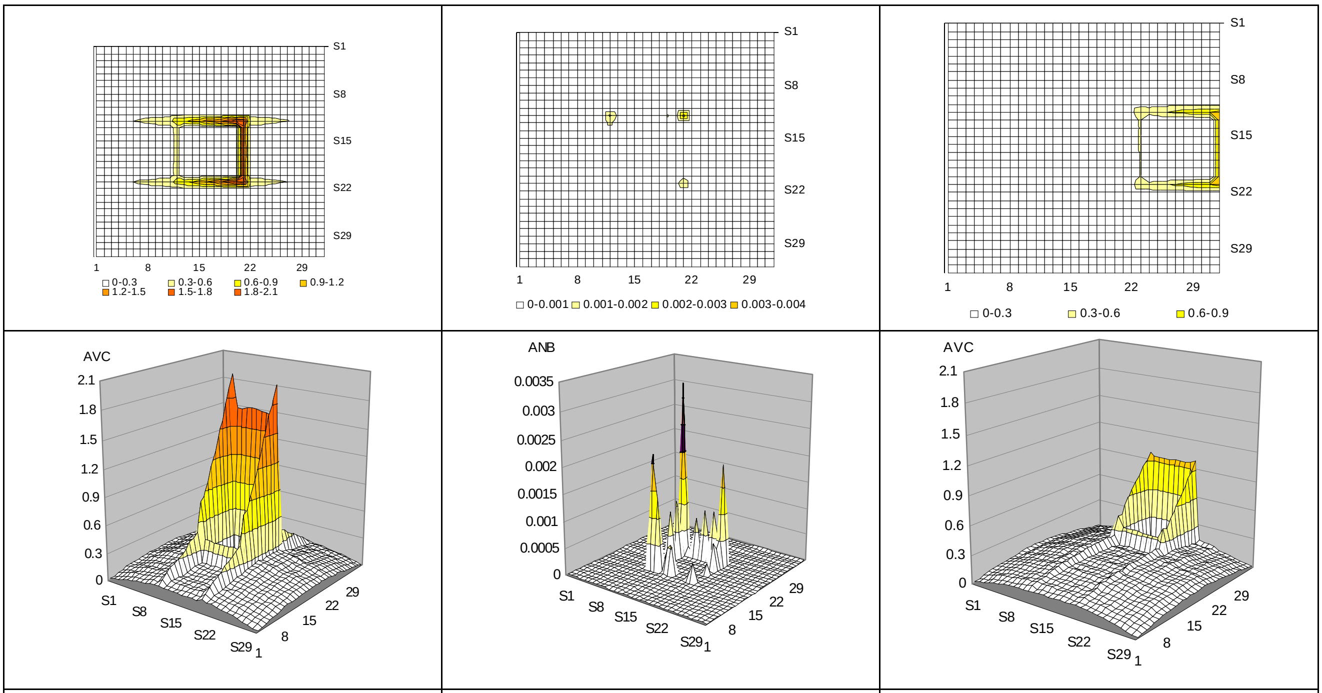


(d) Latency = 59.54 cycles

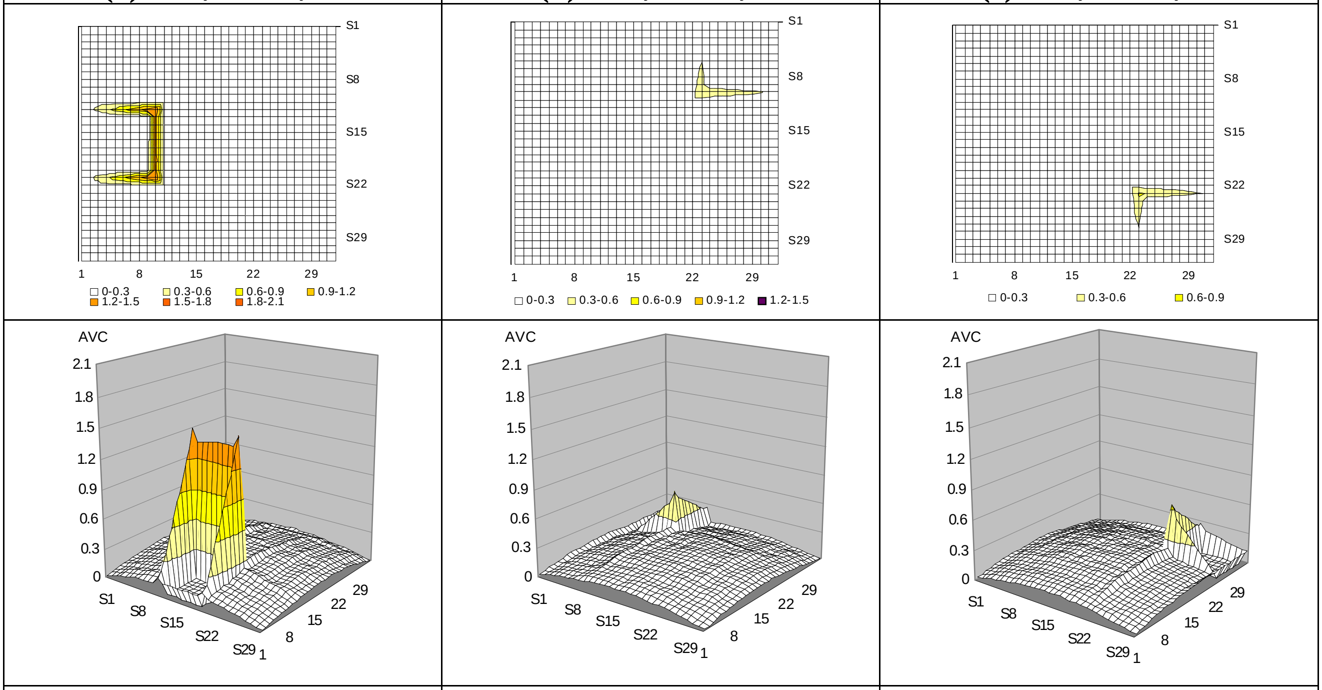
(e) Latency = 64.11 cycles

(f) Latency = 59.37 cycles

Fig. 4. Traffic level (AVC) for various centered fault regions. (a) is a fault free network in normal scale (b) is a fault free network in detailed scale. (c),(d),(e), and (f) are 1 ×1, 4 × 4, 1 × 8, and 8 × 1 regions respectively



(a) Latency = 66.00 cycles **(b)** Latency = 66.00 cycles **(c)** Latency = 60.00 cycles



(d) Latency = 61.89 cycles **(e)** Latency = 56.88 cycles **(f)** Latency = 57.21 cycles

Fig. 5. Traffic level for a 8 × 8 fault region in various positions. (a) is a centered fault region (b) is a centered fault region but plots the ANB measure. (c),(d),(e), and (f) are east, west, north-east, and north-west fault regions respectively