Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

Congestion aware routing algorithm for mesh network-on-chip platform

Arash Dana[1] and Nastaran Salehi[2]
[1]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[2]Islamic Azad University, Science and Research Branch, Tehran, Iran
arash.dana@iauctb.ac.ir, nastaran_salehi60@yahoo.com

Abstract

The routing algorithm plays an important role in the performance of the network on chip. In this paper we propose an adaptive routing, which uses a fuzzy controller to combine two congestion metrics free buffers and crossbar demand. The objective of the proposed routing algorithm is, choosing channel that has more free-slots input buffer beyond adjacent routers and the less number of active requester for a given output port. Simulation results show that the proposed method applied to odd-even routing algorithm can effectively improves average delay and throughput to meet load balance requirement and avoid hotspot with low hardware overhead.

Keywords. Network on Chip, System on Chip, Fuzzy Controller, Congestion, Routing.

Introduction

System on chip (SoC) grows in complexity with the advance of semiconductor technology enabling integration of dozen of cores on a chip. However as the number of component in a single chip increases at a high speed, low energy and high performance communication between them become the bottleneck (Kumar *et al.,* 2002). Network on chip (NoC) design paradigm has been proposed as a viable communication solution for integrating large number of processing cores into a single chip (Kumar *et al.,* 2002; Benini & Micheli, 2002). In NoCs, routing algorithms are used to determine the path of a packet from the source to the destination. These algorithms are classified as deterministic and adaptive routing. Although the implementation of deterministic routings is simple but they are not able to balance load in non-uniform and bursty traffic.

However, adaptive routing requires network path diversity between the source and the destination nodes to facilitate load balance. The availability of network path diversity depends on the topology of the network, the traffic pattern, and whether the non-minimal routes are allowed (Gratz *et al.,* 2008). Adaptive routing can effectively avoid hotspots or faulty components and can reduce the possibility of packets being continuously blocked. An Adaptive routing algorithm gives better communication performance like packet latency and throughput than a deterministic routing algorithm, especially at higher network loads and it makes a packet avoid passing from a congested links. Adaptive routing contains routing and selection functions. The routing functions supply a set of output channels based on the positions of current and destination nodes and the selection function chooses an output channel from the set of channels given by the routing function (Duato *et al.,* 2002).

The selection function can be classified as either congestion-oblivious or congestion-aware schemes (Ascia *et al.,* 2008). In congestion-oblivious algorithms, routing decision is independent of the congestion condition but in congestion-oblivious methods, the selection is usually performed using the congestion status of the network (Duato *et al.,* 2002). Many routing algorithms proposed to determine the congested areas to route packets through the less congested areas to choose an output channel. The major part of the state of the art proposals for adaptive routing algorithms considers local traffic monitoring and each router analysis their immediate neighbors considering congestion metric to choose the output port. This approach presents a limited view of the network traffic which can cause to unbalanced distribution of traffic load.

Gratz *et al.* (2008) proposed a routing technique where congestion information is taken in parts of the network beyond adjacent routers. This information is propagated across a specific network to have a global view of congestion. Duato *et al.* (2002) introduced a static routing algorithm for two-dimensional meshes which is called XY. In this routing algorithm, each packet first travels along the X and the Y direction to reach the destination. An adaptive routing algorithm named turn-model is introduced by Class and Ni (1994) and based on which another adaptive routing algorithm called 0dd-even is proposed by Chiu (2000). To avoid deadlock, Odd-Even method restricts the position that turns are allowed in the mesh topology.

Another algorithm called DyAD is introduced by Hu and Marculescu (2004) which is a combination of a static routing algorithm called OE-fix and an adaptive routing algorithm based on the odd-even turn algorithm. Depending on the congestion condition of the network corresponding to the input buffers occupation, one of the routing algorithms is selected. Congestion flags information is exchanged between neighbor routers. If the router neighbors are not congested, the DyAD router work on deterministic mode, otherwise the adaptive mode is used.

An adaptive deadlock free routing algorithm called

Dynamic XY (DyXY) has been proposed by Li *et al.* (2006). This algorithm is based on the static XY algorithm, a packet is sent either to the X or Y direction depending on the congestion condition.

Ascia *et al.* (2008) have presented a novel selection strategy called NoP that can be coupled with any adaptive routing algorithm. The proposed selection strategy is based on the concept of neighbours-on-path the aim of which is to exploit the situation of indecision occurring when the routing function returns several acceptable output channels. Wu *et al.* (2006) the paper investigates the impact of input selections and presents a novel contention-aware input selection (CAIS) technique for NoC that improves the routing efficiency. When there are contentions of multiple input channels competing for the same output channel, CAIS decides which input channel obtain the access depending on the contention level of the upstream switches, which in turn removes possible network congestion. An application specific routing algorithm named ASPRA has been proposed by Palesi *et al.* (2006). ASPRA exploits communication information to maximize the adaptivity while ensuring deadlock free routing for an application.

Schweibert and Bell (2002) present a detailed study of various functions for several fully adaptive worm hole routing for 2D meshes. The obtained results show that the choice of selection function has a significant effect on the average message latency and saturation behaviour. Feng and Shin (1997) show that in addition to adaptively, the selection function greatly affects network performance under various different traffic patterns. Ye *et al.* (2004) present a contention-look-ahead on chip routing. it is a nonminimal routing in the sense that based on the value of two delay penalty indices, the router chooses whether to send the packet toward a profitable route(minimal route) or a misroute (nonminimal route).

Ebrahimi *et al.* (2011) proposed an agent-based Network-on-Chip to determine the congested areas in the network and route packets through the less congested areas based on the local/non-local congestion information. A novel routing algorithm, named Balanced Adaptive Routing Protocol (BARP) is proposed by Lotfi-Kamran *et al.* (2008) for avoiding congestion and it distributes input packets of a router among all its shortest path output ports. Tedesco *et al.* (2010) propose an adaptive source routing algorithm where the path between source and target may be modified due to the congestion events. This method uses information collected on the source-target path to execute adaptive source routing. Comparing the present work to the state of art, the following differences can be pointed out: (i) considering composite important congestion metrics (ii) using fuzzy controller for optimal routing decisions.

A novel selection strategy for avoiding congested areas using a fuzzy-based routing decision is proposed by Salehi *et al.* (2010) that can be used with any adaptive routing algorithm. The objective of the proposed selection strategy is to choose a channel that has more free slots input buffer and lower power consumption. Salehi *et al.* (2011) proposed a novel fully adaptive routing algorithm for avoiding congested areas using a fuzzy-based routing decision.

Each of the metrics has strengths and weakness for congestion avoidance. In this paper we propose a fuzzy based congestion control approach to address the congestion control problem which can achieve by using a fuzzy controller to combine two congestion metrics free buffers and crossbar demand respectively. The objective of the proposed routing algorithm is, choosing channel that has more free-slots input buffer beyond adjacent routers in a path from the source to the destination and the less number of active requester for a given output port.

Proposed routing algorithm

The objective of the proposed routing algorithm is to avoid the problem of congestion and hotspots in Network on Chip. This is achieved by using fuzzy controller to consider composite congestion metrics.

Fuzzy controller

Fuzzy logic control system is rule-based system (Driankov *et al.,* 1993) in which a set of so-called fuzzy rules represents a control decision mechanism to adjust the effects of certain causes that come from the system. The aim of the fuzzy control system is normally to substitute for or replace a skilled human operator with a fuzzy rule-based system. Fuzzy logic allows us to take into account the continuous character of imprecise information and to avoid arbitrary rigid boundaries. It also has the advantage of establishing an interface between symbolic and numeric data. The basic idea of fuzzy control is to make use of expert knowledge and experience to build a rule base with linguistic if-then rules. Different from other control methods, fuzzy control does not involve complex mathematical operations and models of systems. The fuzzy logic uses linguistic descriptions to define the relationship between the input information and the output action.

In a network, the various metrics like collisions, traffic level, buffer occupancy, etc. need to be considered for congestion avoiding routing algorithm. It is not enough if only one constraint is considered. Multi-constrained routing is a NP-complete problem and does not have a polynomial solution. This is because of the complex

Fig. 1. The fuzzy controler

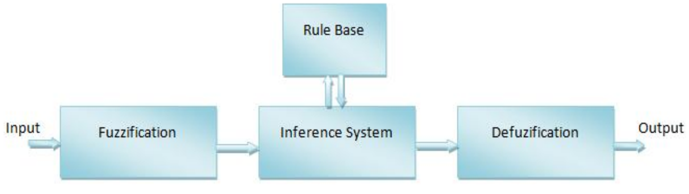


Fig. 2. Router signals for calculating free buffers value

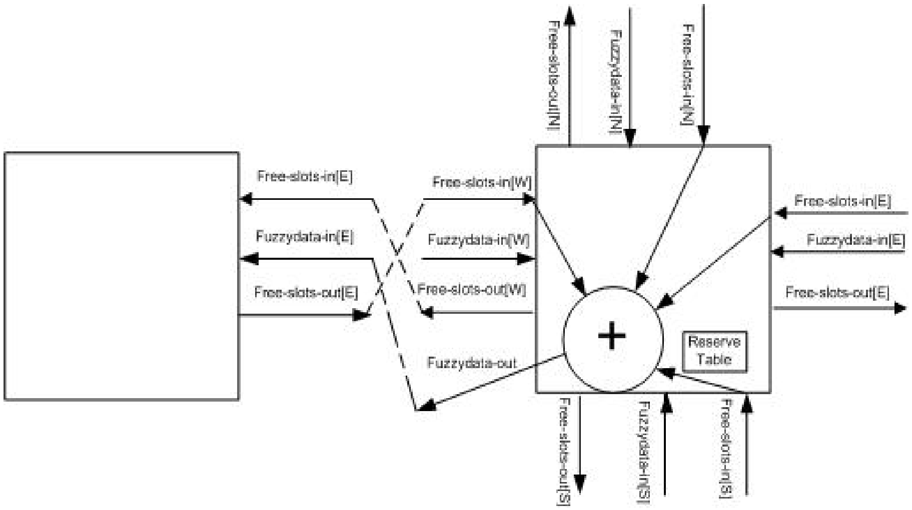
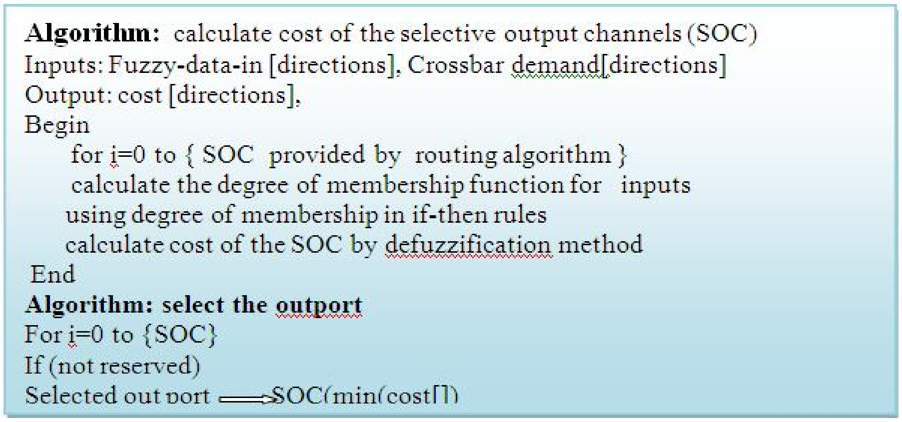


Fig.3. Pseudo code for the fuzzy selection algorithm



```
Algorithm: calculate cost of the selective output channels (SOC)
Inputs: Fuzzy-data-in [directions], Crossbar demand[directions]
Output: cost [directions].
Begin
    for i=0 to { SOC  provided by  routing algorithm }
    calculate the degree of membership function for  inputs
    using degree of membership in if-then rules
    calculate cost of the SOC by defuzzification method
 End
Algorithm: select the outport
For i=0 to {SOC}
If (not reserved)
Selected out port ⟹ SOC(min(cost[])
```

relationship existing between the different constraints. It is required to use various heuristics and soft computing techniques to solve them. A Fuzzy system is best suited in making optimal routing decisions in a network involving multiple constraints and multiple objectives.

The fuzzy logic controller is composed of three main components: Fuzzification, Fuzzy Inference mechanisim and Defuzzification. Fig.1 shows the components of fuzzy logic controller.First we have to fuzify the inputs or create membership values and put them into the fuzzy sets which are normalized in the range of (0,1). The Fuzzification transforms the crisp value of the input variable into the fuzzy sets. Many types of curves can be used but triangular or trapezodical shaped membership functions are the most common.

The inference mechanisim applies reasoning to compute fuzzy output. Mamdani's method is the most commonly used in applications, due to its simple structure of 'min-max' operations. It consists of a number of conditional IF THEN rules that describe the system behaviour and determine which output ranges are used. Defuzzification is the process of producing a quantifiable result and converts the fuzzy control action into a crisp value.

The defuzzification interface converts the fuzzy conclusions of the inference mechanism reaches to a numeric value. Center-of-Gravity is the one of the most important method in defuzification which finds the geometrical centre. Another method for defuuzification is Mean-Of-Maxima which finds the value with the maximum membership degree according to the fuzzy membership function.

Proposed fuzzy Controller

First: We consider free buffers and crossbar demand as inputs and cost of the link as output. Linguistic values for free buffers are {low, middle, high} and crossbar demand is {small, medium, big}. We adopt triangle shape for the fuzzy sets while the fuzzy terms in conclusions are singletons**.** Each quantifications of the variable is assigned a membership function**.**

Free buffers: Buffer count indicates the amount of backpressure that the input port at the downstream node is experiencing. For the better congestion avoidance we consider free buffer beyond adjacent routers. For gathering this value we use the score calculation method which is described by Ascia *et al.* (2008) as shown in Fig. 2 which is described later. This parameter is computed by referring the available buffer of the next node and the neighboring nodes of the next node with Odd-Even turn model.

Crossbar demand: This metric was introduced by Gratz *et al.* (2008) which measures the number of active requester for a given output port. Crossbar demand captures the actual amount of channel multiplexing a new packet is likely to experience. Multiple concurrent requests for an output port indicate a convergent traffic pattern.

*Second:*The inference mechanism applies a predetermined set of linguistic rules in the rule base and produces the fuzzy sets of the output linguistic variable. For the inference mechanism, the max-min method is adopted. Table 1 shows the rules for fuzzy controller.

Finally: In the defuzzification interface, the most popular

Table 1. Rule base for fuzzy based approach for adaptive routing

Crossbar Buffer	Low	Middle	High
Small	Medium	High	Very high
Medium	Low	Medium	High
Big	Very low	Low	Medium

Center of gravity method is used to produce a real number.

Fuzzy routing algorithm

We introduce some signals required between a node and its adjacent neighbors, as shown in Fig. 2. For each direction d∈ {North, South, East, West} we indicate:

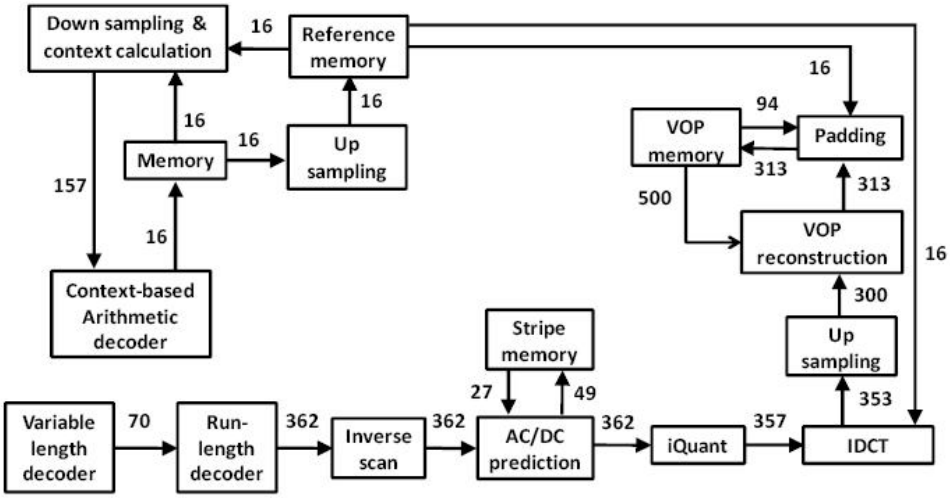
Free-slots-in[d]: number of free buffers slots available at the input buffers of the adjacent neighbor along the direction d;

Free-slots-out[d]: number of free buffers slots available at the input buffers of the current node along the direction d;

Fuzzydata-in[d]: the data collected from the adjacent neighbor along the direction d.

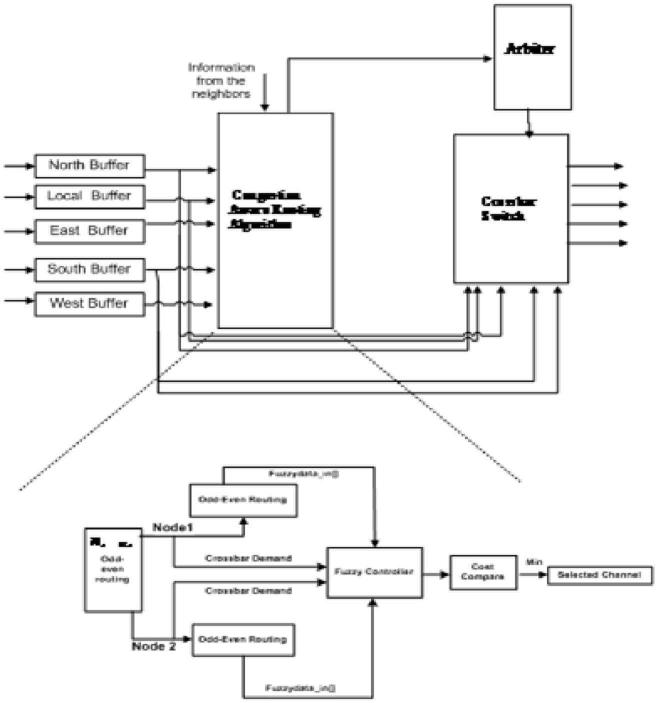
Fuzzydata-out[d]: data collected by the current node and gathering the free slots input buffers from its adjacent nodes.

Fig.5. VOPD block diagram with communication bandwidth annotated (in MB/s)



The proposed congestion-aware adaptive routing algorithm adopts Odd-even turn model to solve the deadlock problem (Chiu, 2000) and it can use for all proposed adaptive routing algorithms.The odd-even turn model prohibits some turns for elimination deadlock. For providing free buffers value as input for fuzzy controller, the neighbors-on-path method is used (Ascia *et al.,* 2008). In this method, odd-even routing algorithm provided two selective output channels then these selective output channel considered as source nodes and again odd-even routing provide second selective output channels(neighbors of adjacent routers).

Fig. 4.Schematic of a router for a fuzzy routing



By finding neighbors and gathering available free slots buffer (Fig.2) and considering this parameter as one of the input parameter for fuzzy controller, helps the proposed fuzzy routing algorithm does not try to send a packet to a network path that is in the process of getting congested few hops after, but not yet propagated to the current switch. By considering this parameter and crossbar demand as two input metrics and cost calculation by the fuzzy controller for each selective output channels, the output port with the minimum cost is selected. Fig.3 shows the pseudo code of the fuzzy routing algorithm.

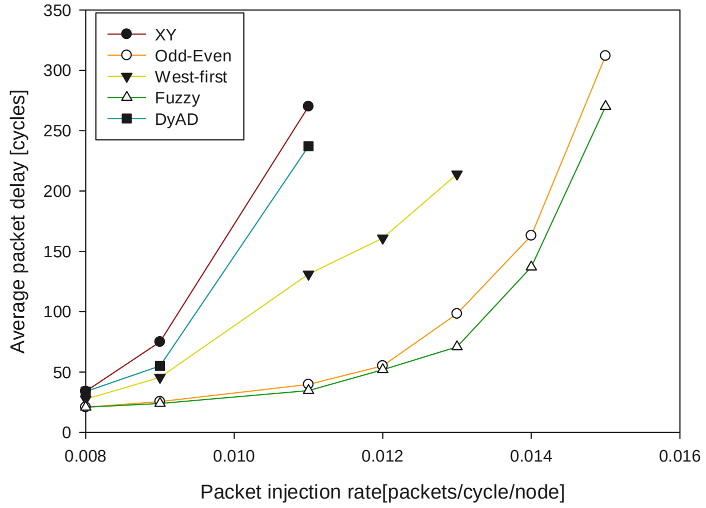
NoC router

Fig.4 shows a schematic of a router for a mesh-based network on chip implementing our proposed routing algorithm. As shown, the router consists of a set of input buffers for each direction and an input buffer for the traffic locally generated. A routing algorithm is used when a routing has to be made, when the header reaches the input buffer. The output of the routing algorithm block is the selected output channel is used by the arbiter to setup the crossbar interconnection that connects input to the proper output directions.

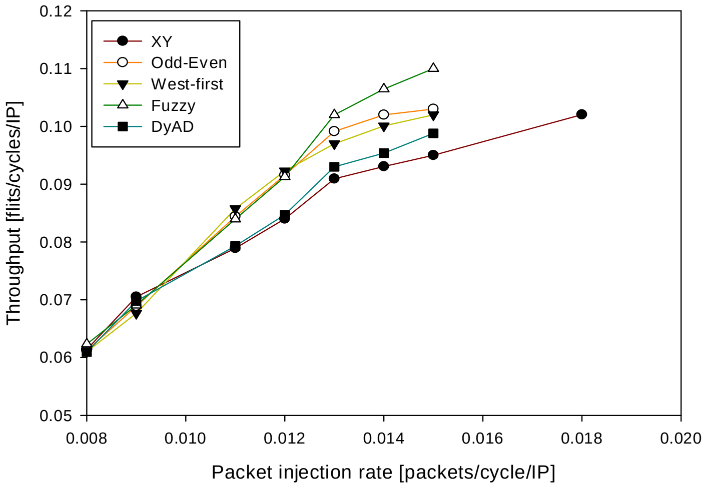
The second block describes how the fuzzy controller works. Considering a 2D mesh topology and a minimal routing algorithm there can be a maximum of two selective channels for each routing block. Then for each selective channel the odd-even routing provides new selective channels which consider the previous selective channels as the source nodes. After that with use of a counter for calculating crossbar demand value for each first output channels and calculated free buffers, the fuzzy controller for each two sets calculate the cost value and finally the channel with the lowest cost is selected for output. Figure shows the pseudo-code of the algorithm.

Fig.6 Simulation results for transposed traffic

a) Average delay variation



b) Average throughput variation



c) Power consumption

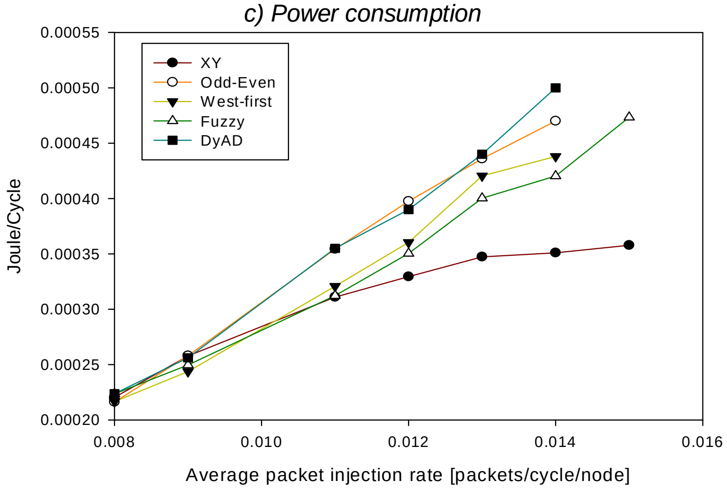
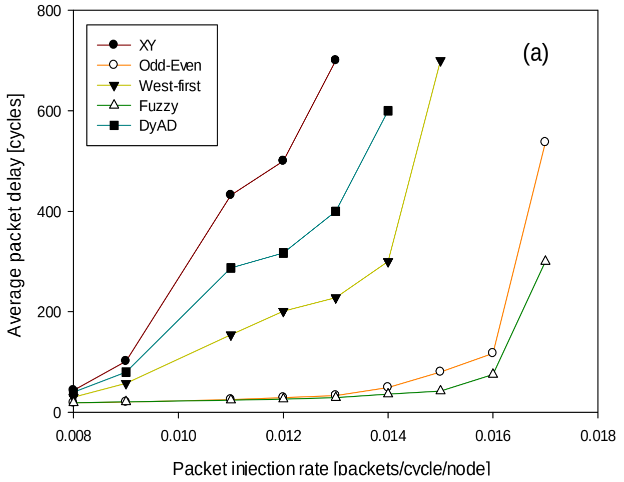
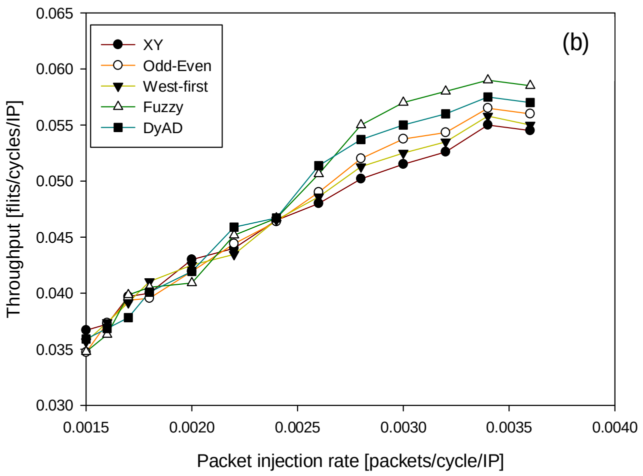


Fig.7. Simulation results for hotspot traffic

a) Average Delay variation



b) Average Throughput variation



c) Power consumption

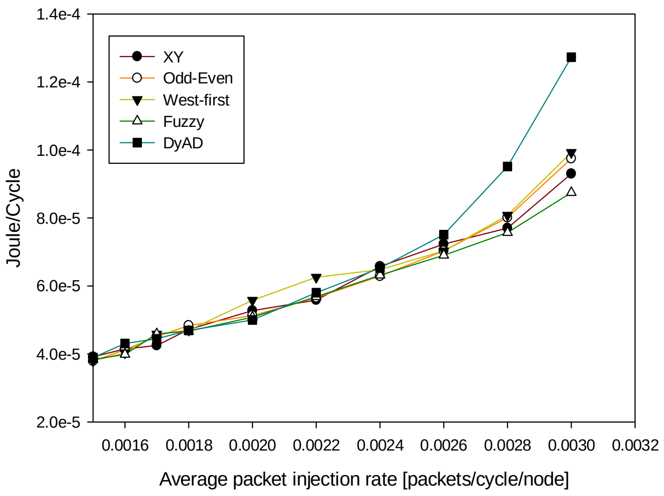
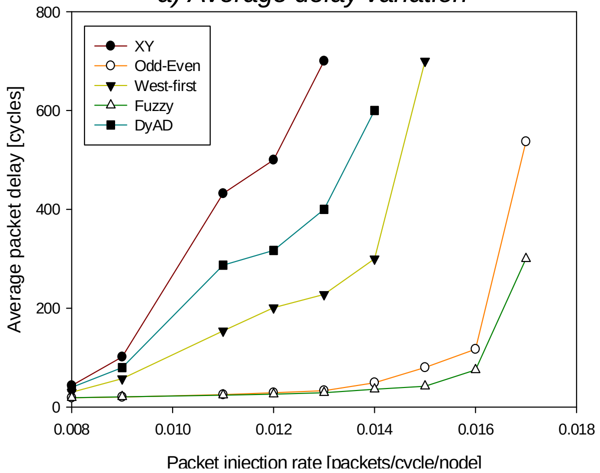
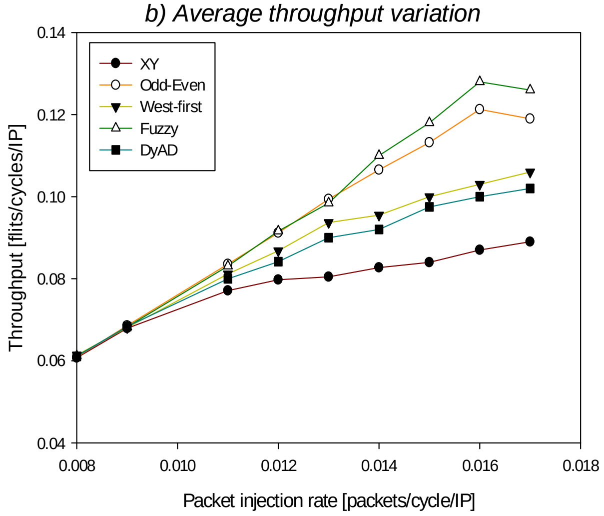


Fig.8. Simulation results for bit-reversal

a) Average delay variation



b) Average throughput variation



c) Power consumption

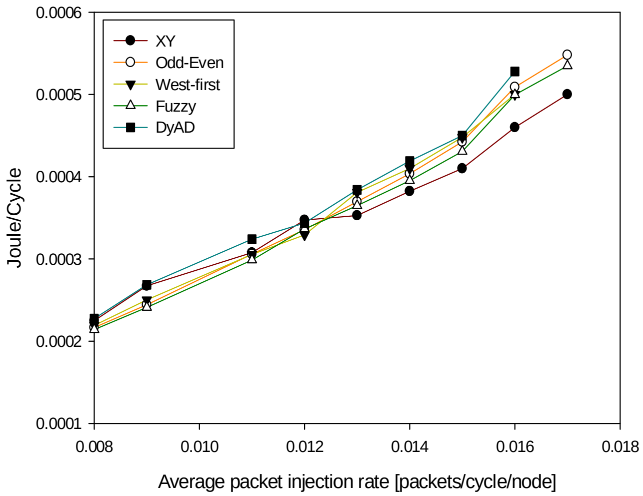
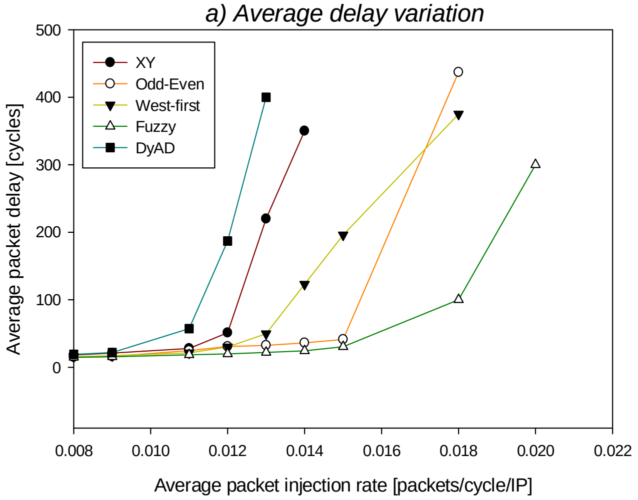
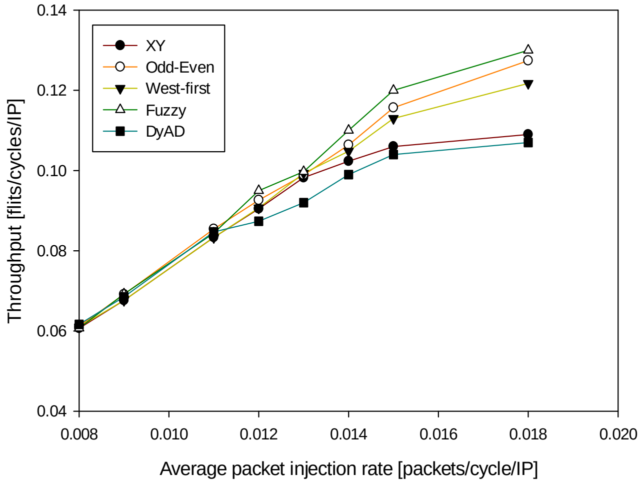


Fig.9 .Simulation results for shuffle traffic

a) Average delay variation



b) Average throughput variation



c) Power consumption

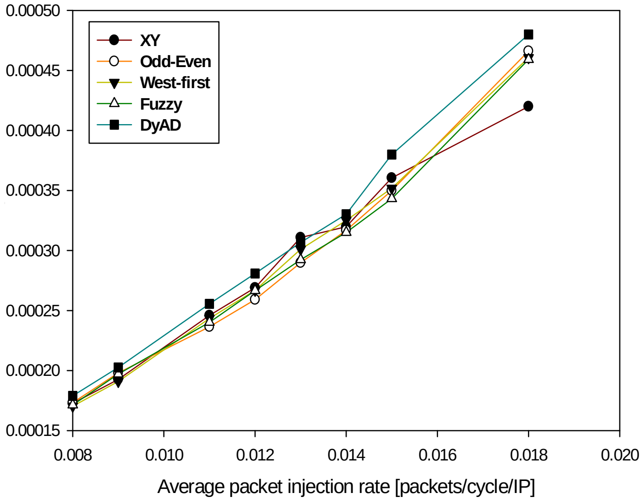
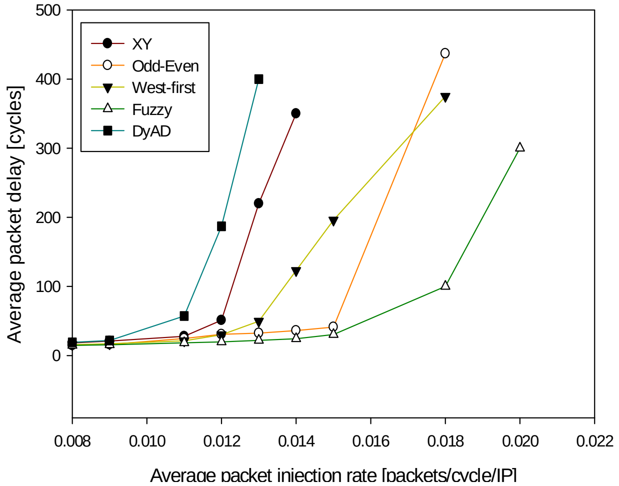
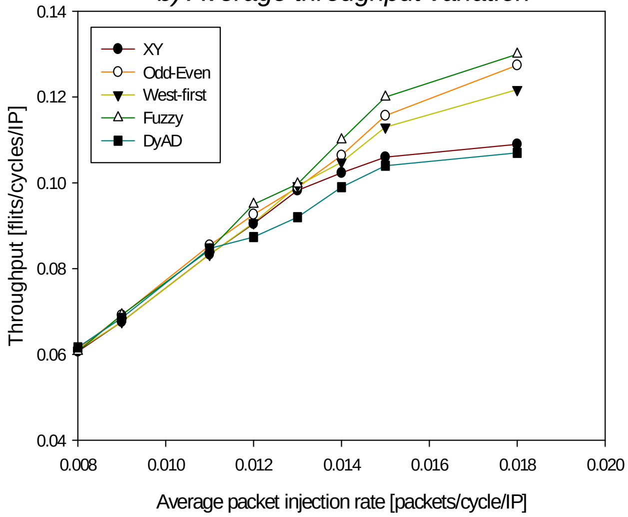


Fig.10. Simulation results for VOPD

a) Average Delay variation



b) Average throughput variation



Experimental results

Simulation Environment

We evaluate congestion aware routing by extending Noxim (Sourceforge, 2008) (an open source SystemC simulator of a mesh-based NoC) using four synthetic traffic patterns. Data width was set to 32 bits. Simulation were performed on a 8*8 mesh NoC and each node has 6 slots buffer and generated 8 flit packets with an exponentional distribution. Simulation runs for 1000 cycle for a warm-up and executes for 20000 cycles.

Traffic scenarios

We evaluate the proposed approach on both synthetic and real traffic scenarios. As synthetic we consider transpose, bit-reversal, hotspot and shuffle. For each traffic, average delay and throughput and power consumption with various packet injection rate has been evaluated. Finally, for a realistic communication we consider Video Object Plane Decoder(VOPD) mapped onto 3*4 mesh topology (Vander & Jaspers, 2002). Fig.5 shows the VOPD block diagram with communication bandwidth.

Evaluation metrics

As performance metrics, we choose throughput and delay. Throughput defines as follows:

$$Throughput = \frac{Total\ received\ flits}{Number\ of\ nodes \times Total\ cycles} \qquad (1)$$

Where Total received flits refers to the number of whole flits that arrive at their destination, Number of nodes is the number of nodes, and Total cycles is the number of clock cycles between the time which first message generation and the last message reception. Delay is defined as the time that header flit injects into the network at the source node and tail flit receive at the destination node.

We use average delay D as follows:

$$Delay = \frac{1}{N}\sum_{i=1}^{N} Di \qquad (2)$$

Where N is the total number of messages reaching to destination and is the delay of message i.

Fig.6 shows the simulation results for transposed traffic. In this traffic, a node (i,j) in mesh network topology only sends packets to a node (N-1-i, N-1-j), where N is the size of the mesh topology. In transposed traffic (Fig.6), XY performs weakly because of its determinisim in distributing packets. Under nonsaturated traffic conditions, the Fuzzy routing schemes gives a 12 percent to 88 percent improvement in average delay as compared to other routing algorithms.

In Hotspot traffic, four nodes located at the center of the mesh, [(4,4),(4,5),(5,4),(5,5)] with 20 persent hotspot traffic are considered as hotspot nodes which receive hotspot traffic in addition to regular uniform traffic. As we can seen being able to route packets on the basis of congestion information received from neighbors allows fuzzy to have a higher saturation point with better performance under nonsaturated traffic (Fig.7).

Fig.8 shows the simulation results for bit-reversal traffic. In this traffic a node sends message to the node with its reversal coordinates. The Fuzzy routing schemes gives a 50 percent to 94 percent improvement in average delay as compared to other routing algorithms.

Fig. 9. shows the simulation results for shuffle traffic. The Fuzzy routing schemes gives a 26 Percent to 95 percent improvement in average delay as compared to other routing algorithms.

Table 2 shows the percent improvement of fuzzy routing algorithm over XY, OE, West-first DyAD. In terms of average delay, on average, fuzzy outperforms XY, odd-even, west-first and DyAD by 91 percent, 31 percent, 81 percent and 93 percent respectively.

For power dissipation, we show the average Joule/Cycle consumed for each traffic scenario at different packet injection rate. As the injection rate increases, the average power dissipation increases. We can observe that fuzzy average power dissipation trend does not increase from other router implementation and in some cases it has a little increase as compared to XY and still under the adaptive routing. In order to analyse the power, we use the method which is used in Noxim simulator (Sourceforge, 2008).

In this method the energy dissipated in router calculated by running Synopsis Design power on the gate-level netlist of the router. The average energy dissipated by a flit for a hop switch was estimated for XY, OE, DyAD, West-first and fuzzy. It assumed the tile size to be 2mm*2mm and the tiles were arranged in a regular fashion on the floorplan. The load wire capacitance was set to 0.50 ƒF per micron, so considering an average of 25 percent switching activity the amount of energy consumed by a flit for a hop interconnect is 0.384 nJ.

Hardware overhead

To evaluate the area overhead of the proposed algorithm compared to other routing algorithms, we synthesized the VHDL reference model with Synopsys Design Compiler using a standard CMOS library. For all switches, the data width was set to 32 bits and each channel had a buffer size of six flits. The results show the area overhead of switches comparing the proposed routing with XY, Odd-Even, West-first, DyAD and proposed with 11.82%, 5.6%, 5.3% and 4.3% overhead respectively.

Conclusion

In this paper we propose a fuzzy based congestion control approach to address the congestion control problem which can achieve by using a fuzzy controller to combine two congestion metrics, free buffers and crossbar demand respectively. The fuzzy system was able to match the best performance of the composite congestion metrics under different loads. Simulation results show that the proposed method applied to odd-even routing algorithm can effectively improve average delay and throughput to meet load balance requirement and avoid hotspot with low hardware overhead.

Acknowledgement

The authors wish to acknowledge the Central Tehran Branch, Islamic Azad University, Tehran,Iran for the financial support during the course of this research.

Table 2. Improvement in average delay of fuzzy as compared to XY, OE, West-first, DyAD

Traffic	Fuzzy delay improvement (%)			
	XY DyAD	OE		West-first
Transposed	87	12	74	88
Hotspot	94	36	89	95
Bit-reversal	94	50	90	94
Shuffle	92	26	73	95
Average improvement	91	31	81	93

References

1. Ascia G, Catania V, Palesi M and Patti D (2008) Implementation and analysis of a new selection strategy for adaptive routing in network-on-chip. *IEEE Transac. Comp.* 57(6), 809-820.
2. Benini L and De Micheli G (2002) Network on chips: A new SoC paradigm. *IEEE Computers.* 35(1), 70-78.
3. Class CJ and Ni LM (1994) The turn model for adaptive routing. *J. ACM.* 41, 874-902.
4. Chiu GM (2000) The odd-even turn model for adaptive routing. *IEEE Transact. Parallel & Distributed Sys.* 11, 729-738.
5. Duato J, Yalamanchili Y and Ni L (2002) Interconnection networks: An engineering approach, Morgan Kaufmann.
6. Driankov D, Hellendoorn H and Reinfrank M (1993) An introduction to fuzzy control, Springer-Verlag, Berlin, NY.
7. Ebrahimi M, Daneshtalab M, Liljeberg P, Plosila J and Tenhunen H (2011) Efficient congestion-aware selection method for on-chip networks. *IEEE workshop on Reconfigurable Commun.-centric Sys. Chip (ReCoSoC).*
8. Feng WC and Shin KG (1997) Impact of selection functions on routing algorithm performance in multicomputer networks. *Supercomputing*. pp: 132-139.
9. Kumar S, Jantsch A , Soininen JP, Millberg M, Oberg J, Tiensyria K and Hemani A (2002) A network on chip architecture and design methodology.*ISVLSI*. pp: 117-122.
10. Gratz P, Grot B and Keckler SW (2008) Regional congestion awareness of load balance in network-on-chips. *Proc. Intl. Symp. High Performance Comput. Architec.* pp: 203-214.
11. Hu JC and Marculescu R (2004) DyAD smart routing for networks-on-chip. *Design Automation Conf.* pp: 260-263.
12. Li M, Zeng QA and Jone WB (2006) DyXY a proximity congestion-aware deadlock-free dynamic routing method for network on chip. *Design Automation Conf.* pp: 849-852.
13. Lotfi-Kamran P, Daneshtalab M, Navabi Z and Lucas C (2008) BARP a dynamic routing protocol for balanced distribution of traffic in NoCs to avoid

congestion. *ACM/IEEE design automation & test in Europe Conf. (DATE)*. pp: 1408-1413.

14. Noxim (2008) Network-on-chip simulator, http://sourceforge.net/projects/noxim
15. Palesi M, Holsmark R, Kumar S and Catania V (2006) A methodology for design of application specific deadlock-free routing algorithms for NoC systems hardware-software co design and system. *Synthesis.* pp: 142-147.
16. Palesi M, Holsmark R, Kumar S and Catania V (2009) Application specific routing algorithms for networks on chip. IEEE Transact. *Parallel & Distributed Sys.* 20(3), 316-330.
17. Schweibert L and Bell R (2002) Performance tunning of adaptive wormhole routing through Selection Function Choice. *J. Parallel & Dstributed Comput.* 62(7), 1121-1141.
18. Salehi N, Khademzadeh A and Dana A (2010) Power distribution in NoCs through a fuzzy based selection Strategy for Adaptive Routing 13th Euromicro Conference on Digital System Design: Architectures, *Methods & Tools*. pp: 45-52.
19. Salehi N, Khademzadeh A and Dana A (2011) Minimal fully adaptive fuzzy-based routing algorithm for Networks-on-Chip *IEICE Elect. Express*. 8(13), 1102-1108.
20. Tedesco L, Rosa Th, Clermidy F, Calazans N and Moreas F (2010) Implementation and evaluation of a congestion aware routing algorithm for networks-on-chip. Symposium on integrated circuits and system design. pp: 91-96.
21. Vander Tol EB and Jaspers EGT (2002) Mapping of MPEG-4 decoding on a flexible architecture platform SPIE. pp: 1-13.
22. Wu D, Al-hashimi BM and Schmits MT (2006) Improving routing efficiency for network-on-chip through contention-aware input selection Asia and South Pacific. *Design Automation*. pp: 36-41.
23. Ye TT, Benini L and Micheli GD (2004) Packetization and routing analysis of on-chip multiprocessor Networks. *J.Sys. Architectures*. 50, 81-104.