Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8588646, 11 pages
https://doi.org/10.1155/2021/8588646

Research Article

Congestion-Aware Routing Algorithm for NoC Using Data Packets

Khurshid Ahmad,[1] Muhammad Athar Javed Sethi,[1] Rehmat Ullah,[1] Imran Ahmed,[2] Amjad Ullah,[3] Naveed Jan,[4] and Ghulam Mohammad Karami[5]

[1] *Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar 25000, Pakistan*
[2] *Center of Excellence in IT, Institute of Management Sciences, Peshawar 25000, Pakistan*
[3] *Department of Electrical Engineering, University of Engineering and Technology, Peshawar 25000, Pakistan*
[4] *Department of Information Engineering Technology, University of Technology, Nowshera 24100, Pakistan*
[5] *SMEC International Pvt. Limited, Kabul 1007, Afghanistan*

Correspondence should be addressed to Ghulam Mohammad Karami; ghulam.karami@smec.com

Received 17 May 2021; Revised 26 July 2021; Accepted 10 August 2021; Published 19 August 2021

Academic Editor: Sungchang Lee

Network on Chip (NoC) is a communication framework for the Multiprocessor System on Chip (MPSoC). It is a router-based communication system. In NoC architecture, nodes of MPSoC are communicating through the network. Different routing algorithms have been developed by researchers, e.g., XY, intermittent XY, DyAD, and DyXY. The main problems in these algorithms are congestion and faults. Congestion and faults cause delay, which degrades the performance of NoC. A congestion-aware algorithm is used for the distribution of traffic over NoC and for the avoidance of congestion. In this paper, a congestion-aware routing algorithm is proposed. The algorithm works by sending congestion information in the data packet. The algorithm is implemented on a 4 × 4 mesh NoC using FPGA. The proposed algorithm decreases latency, increases throughput, and uses less bandwidth in sharing congestion information between routers in comparison to the existing congestion-aware routing algorithms.

1. Introduction

Semiconductors revolutionized the microelectronics world in all aspects that include the military, computers, medical field, telecommunication, and aerospace. The increasingly lower cost per transistor increases the transistor's integration up to 20 million transistors per chip because of the industrial revolution that leads to System on Chip (SoC) in the application area of military, computer peripheral, DSP, communication, and multiprocessor. SoC's functional unit connectivity is a complex task because it is more important for the system's overall performance [1, 2].

SoC is a bus-based system, and because of nonscalability, it cannot fulfill the requirement of performance. In SoC, the bandwidth of the bus is shared between devices, and it is insufficient. Based on this reason, in 2000, a new generic interconnection template was proposed, which addresses the performance and scalability requirement for SoC, using a switching network [3, 4]. In this network, data is moved from one terminal to another in a small format called a packet. A packet contains a header, payload, and tail flits. The header comprises destination information. The switching element is called a router. When the router receives the packet, it forwards it to a neighbor router based on its destination information.

Like a computer network, the routing algorithm in NoC can forward data flit (packet in NoC) from source to destination. For the avoidance of congestion and distribution of traffic over the network, the routing algorithms are used in NoC [5]. Different techniques are used by researchers in the development of routing algorithms to control congestion in NoC [6].

A congestion-aware routing algorithm is developed by sending congestion information in data packets across NoC, which controls congestion. The proposed algorithm decreases latency, increases throughput, and uses less bandwidth in sharing congestion information between routers in comparison to existing congestion-aware algorithms.

2. Literature Review

The employed routing algorithm and topology of NoC affect the overall performance. The NoC routing algorithm is the same as the routing algorithm of a computer network with area and cost constraints [5]. The routing algorithm is the set of rules used by the NoC router to traverse a packet from a source to a destination [7–9]. XY, West first, and random routing algorithms are oblivious congestion algorithms [10, 11]. In these algorithms, routing decisions do not depend on the network's status which affects the performance of NoC. Unlike oblivious congestion algorithms, congestion-aware algorithms consider network congestion information when sending data flit from source to destination [12–14].

The DyAD routing algorithm integrates the benefit of adaptive and deterministic routing algorithms [15]. If no congestion is in a neighbor router, then routing is done through deterministic routing. Otherwise, adaptive routing is used for sending flits across NoC. In DyAD, they are flipping between routing schemes based on local congestion information. If the source sends data flit to a destination, the source router selects deterministic routing based on local congestion information. However, if congestion occurs in the neighbor router, then the latency of the flit is increased significantly. In the DyXY routing algorithm, data flit travels through the shortest path [16]. When there are several shortest routes from source to destination, then path selection is based on local congestion information. The DyXY algorithm's weakness is when there are two paths between source and destination (e.g., path1 and path2); then based on congestion information, if there is no congestion in the neighbor router of path1, the DyXY algorithm selects path1 to traverse data flit. However, if in path1 the router next to the neighbor router is congested, then the algorithm chooses the wrong path [17].

The EDXY routing algorithm overcomes the problem of the DyXY algorithm [18]. To share congestion information between routers over row (or column), a congestion wire is used. Through congestion wire, the congestion flags are spread on the path of row (or column), which tells the adjacent row (or column) that this row is congested or not. For implementation, extra hardware is used, which increases area and power consumption. The function of the FADyAD algorithm is the same as DyAD. FADyAD has the same problem as DyAD [19].

In 2014, a congestion controlling routing algorithm based on a dynamic routing table in NoC was developed [20]. After every 10 seconds, routers broadcast their routing table to neighbor routers. At the time of sharing the routing table, no data flits are sent which affects the latency. Another issue is that a more congested node is declared faulty. In regional congestion-aware routing, congestion information is shared between routers when links are idle [21]. When links are not idle, no congestion information is shared between routers. When a network is loaded, and congestion occurs at any router; then, delivery of data flit to a destination is compromised [22].

In MCAR, without a data packet, two types of messages propagate between routers for congestion avoidance [23]. One is normal (0), and the other is congested (1). Each router has four neighbors (north, south, east, and west). For example, when the input port on the north side of the router is crowded from normal, the output port on the north side of the router sends a congested (1) message to the north side router without taking any turns as this causes delay [24].

The congestion-aware, fault-tolerant, and process variation adaptive routing algorithm (CFPA) was developed in 2019 [25]. CFPA avoids congestion based on two routing tables (queuing delay table and propagation delay table) which are maintained in each router of NoC. Overhead in CFPA is periodic broadcasting queuing delay information through queuing delay message (QD_MSGs), and extra memory is required for storage. One other issue is the calculation for finding a path which requires more processing.

In MCAR, when sending a normal or congested message to a neighbor, data flit moving in that direction waits until a normal or congested message is sent. Here is the possibility to attach a normal or congested message to data flits and send it to a neighbor router. In CFPA, without sending QD_MSGs, the delay information is periodically attached to the data flit and sent to a neighbor.

3. Methodology

3.1. Congestion Information Calculation. In [26], a delay model is proposed in which input and output buffer delay and calculation of the router delay are not considered. In the CFPA routing algorithm, a new model of delays is proposed [20]. In this model, they consider propagation delay of the packet and queuing delay in its input port and output port to calculate congestion information. Equation (1) shows the total delay:

$$T_{\mathrm{D}} = P_{\mathrm{D}} + Q_{\mathrm{D}}, \tag{1}$$

where Q_{D} is the queuing delay and P_{D} is the propagation delay. The propagation delay is calculated based on links from the sender router output port to receiving router input port. The queuing delay is the delay that how many packets stay in the sender router output port and an input port of the receiving router.

In the proposed routing algorithm, both propagation delay and queuing delay for congestion information calculation are used. Each router calculates its propagation delay and queuing delay. The total delay is the time in which the flit is staying on the router.

3.2. Propagation Delay. The propagation delay is the time interval taken by the packet when passing through the router. Propagation delay calculation starts when a packet enters into an input port up to when the packet is sent through the output port without waiting in a queue. P_D is the propagation delay of the router, and the components of propagation delay are as below:

(i) Input port delay = P_i

(ii) Delay at RC (routing competition) unit of router = P_{rc}

(iii) Delay at the output port = P_{out}

(iv) When packet moves from the input port to the RC unit and from RC unit to output port, the delays = $2P_{dl}$

Total propagation delay calculated from Equation (2) is the sum of all the delays at the input, output, routing competition unit, and a link between these components:

$$P_D = \sum_{i=0}^{3} (P_i + P_{rc} + P_{out} + 2P_{dl})_i. \quad (2)$$

3.3. Queuing Delay. Queuing delay refers to the time that a packet can take in the queue of the input port and output port of the router. A router has four input ports and four output ports for communication with neighbor routers. The total queuing delay is the sum of all the time that flit can take on the input and output ports. So, the total queuing delays are

$$Q_D = \sum_{i=0}^{3} (Q_{IDi} + Q_{ODi}). \quad (3)$$

In Equation (3), Q_D is the total queuing delay of the router. Q_{IDi} and Q_{ODi} are the delays at their input port and output port, respectively. The components of the queuing delay of a router are illustrated in Figure 1. The total delay is calculated from Equation (1), and the average of T_D is used as congestion information in the proposed routing algorithm. On this base, the router was declared to be congested or not congested. This T_{TD} is calculated on the router and embedded in a flit to share congestion information between the routers.

3.4. Flit Pattern. An NoC flit (flow control unit or flow control digit) is the smallest unit of a packet that carries data from source to destination. The packet is divided into flit. Flit size influences the capability of NoC. When flit size is small, it requires a smaller memory/buffer for storage over a network or on the router. When buffer sizes are small, it shrinks the size of the router, so the size of a network on chip is decreased, but it increases the power consumption. On the other hand, when the flit size is large, the size of the router is increased, which increases the overall area of a network on chip. Flit size for different NoC structures varies, starting from 8 bits to 256 bits. The flit size is 64 bits in the proposed algorithm.

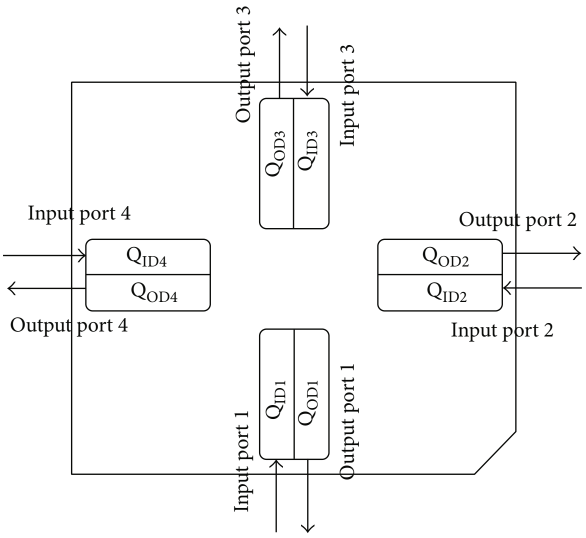

Figure 1: Q delay components.

The communicating data between two nodes is first divided into a packet, and then, the packet contains multiple flits. Flit contains head flit, body flit, and tail flit. Figure 2 shows the structure of the flit. In Figure 2, the first three bytes are head flit, the next 4 bytes are body flit, and the last one is tail flit.

3.5. Head Flit. Head flit contains the source address, destination address, and congestion information. Figure 3 shows a structural organization of the head flit. The first byte is the source address. In the source address, the first four bits show the location of a router on the x-axis, and the last four bits show the location on the y-axis in NoC. The second byte specifies the destination address. The source and the destination address are the same or not changing when the flit is on a route over the network. The last byte of the head flit contains congestion information that is shared between two neighbor routers.

The congestion information is detached and attached when the flit moves from one router to another. The first four bits of congestion information contain the total delay of a current router which is calculated from Equation (1). The last four bits have the conservation status of the router. Each router has four neighbors. The last four bits of congestion information have the congestion status of neighbor routers, and each bit shows the status of one router. Each bit has to state either 0 or 1, 1 for congested and 0 for an uncongested router.

3.6. Body Flit. The body flit has the data, which is sent from source to destination via the network. The size of the body flit is 4 bytes. The data in flit is placed in sequence order.

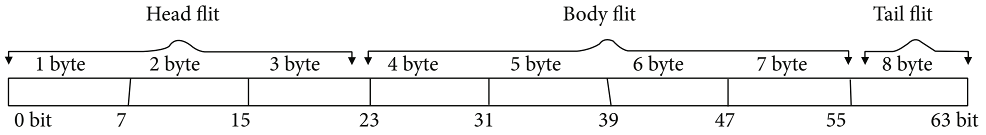

Figure 2: Flit structure.

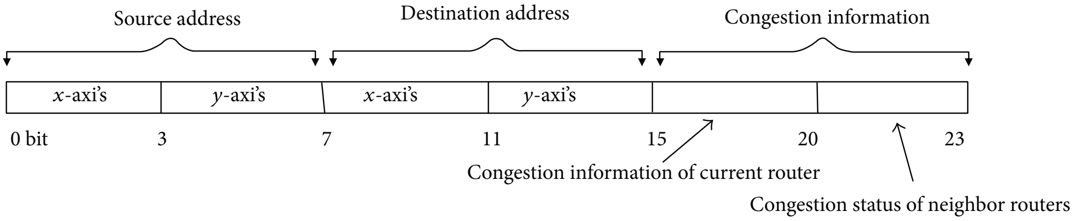

Figure 3: Head flit.

3.7. Tail Flit. Tail flit contains ending information of data and the sequence number of flit. The size of the tail flit is one byte. The seven bits starting from 0 to 6 in tail flit are the sequence number of flit. For example, the data size is 256 bits, which will be sent from source to destination. This data will be sent through eight different flits (flit-0 to flit-7). The flit sequence number shows the flit number on which the data is rearranged at its destination. The maximum number of flit in a single packet is up to 2^7. The size of data in a packet is up to $2^7 \times 4$ bytes.

The last bit of the tail flit shows the ending of a packet. When the bit is on, it means this is the last flit. Otherwise, it is not the last flit of sent data to the destination.

3.8. Proposed Routing Algorithm. The proposed routing algorithm is a congestion-aware routing in which the congestion information is shared between the routers through data packets/flits. The routing decision on sending data from source to destination is taken based on the congestion information of neighbor routers. This algorithm calculates router congestion information according to Equation (1), then shares it with the neighbor router when the packet/flit moves from the current router to the neighbor router. The congestion information is embedded in the head flit and moves to the neighbor router on the route. The router detaches the packet's congestion information and attaches its congestion information and sends the packet through the route.

The proposed routing algorithm for routing decisions relies on the destination address and congestion information table (CITb). CITb stores the congestion information of the neighbor router and next-door neighbor router congestion status. The node connectivity between routers in mesh 4 × 4 topology is the following.

(i) Corner router = 2

(ii) Boundary router = 3

(iii) Centre router = 4

Table 1: Congestion information table (CITb).

S.No.	Neighbor router location	Congestion information of the neighbor router	Congestion status of the next-door neighbor router			
			North	East	West	South
1	North	4 bits	1 bit	1 bit	1 bit	1 bit
2	East	4 bits	1 bit	1 bit	1 bit	1 bit
3	West	4 bits	1 bit	1 bit	1 bit	1 bit
4	South	4 bits	1 bit	1 bit	1 bit	1 bit

Each router stores four neighbors' congestion information and their neighbor congestion status based on the CITb. Let router *R* connect to four routers on his north, south, east, and west. Router *R* CITb contains congestion information of the northside route and the northside router neighbor's congestion status. Router *R* also stores congestion information of all other side routers and the congestion status of their neighbors. Table 1 shows the CITb. The CITb is updated when the packet passes through the router. This routing algorithm has two parts: congestion information sharing and routing decisions.

3.9. Congestion Information Sharing. In this section, congestion information sharing is elaborated. The congestion information sharing algorithm calculates, stores, and shares the congestion information with the neighbor router. The congestion information sharing process cannot require a congestion propagation network. The routers' congestion information is shared when the head flit passes from the router to its neighbor. Figure 4 shows the steps of congestion information sharing for each packet.

The congestion information sharing process relies on the CITb. When the head flit is received on the router, it reads and writes the congestion information byte of the head flit. In reading, it collects congestion information and stores congestion information in CITb with respect to the location of the sender router. In writing, the router calculates congestion information of its own and

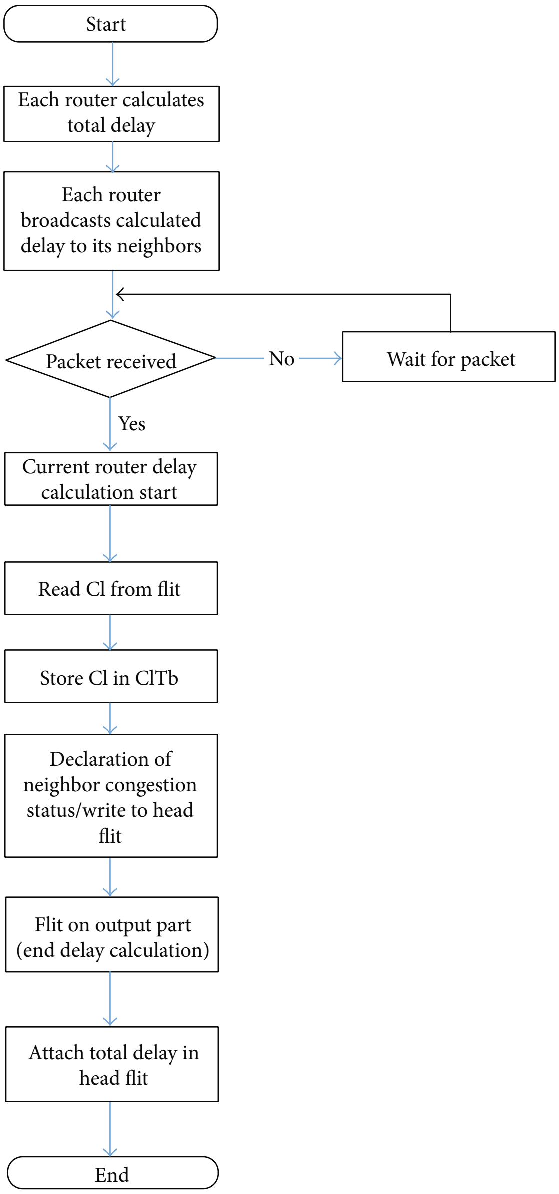
Start
Each router calculates total delay
Each router broadcasts calculated delay to its neighbors
Packet received
No
Wait for packet
Yes
Current router delay calculation start
Read Cl from flit
Store Cl in ClTb
Declaration of neighbor congestion status/write to head flit
Flit on output part (end delay calculation)
Attach total delay in head flit
End

Figure 4: Congestion information sharing steps.

congestion status of its neighbor from stored congestion information in the CITb.

The congestion information of the router and the total delay of the router is calculated through Equation (1). The congestion status of the router means the router is congested or not. The router declares if its neighbor is congested or not and is dependent on the congestion information of that neighbor router which is stored in the CITb. The router will be declared congested when delay/100 > 0.70.

When the router is declared congested, its status is set to 1; otherwise, it is zero. The congestion status of the neighbor router is attached in the congestion information byte of the head flit whenever routing decisions are finalized, and congestion information of the router is attached, when the head flit is on the output port of the router. After the attachment of congestion information byte data in the head flit, the flit is sent to the neighbor router on the route.

3.10. Routing Decision. The routing decision of the proposed routing algorithm depends on the destination address and congestion information stored in the CITb. The routing decision of the proposed routing algorithm has two levels. In the first level, it finds the two alternative paths for flit, which can send the packet to a destination. There are two possible paths to a destination, one on the x-axis and the other on the y-axis. In this level, it compares the destination address of the flit with its address. In comparison, first, it compares the x-axis then the y-axis for finding the two alternative paths. In the second level, it selects the route for flit between these two alternative paths. The selection of paths between these two alternative routes depends on the CITb data. In order to finalize routing decisions, it checks the neighbor router's congestion status of alternative paths from the CITb. Comparison of the congestion status of the neighbor router of both alternative paths has three possibilities.

(i) When both neighbor routers are uncongested, select the route with low delay information of the neighbor router

(ii) When between these two neighbor routers one is congested and another is uncongested, then an uncongested router route is selected for flit routing

(iii) When both neighbor routers are congested, then compare the congestion status of the next-door neighbor of both routes

Comparing the congestion status of next-door neighbor routers has three possibilities.

(i) When both next-door neighbor routers of both routes are congested, select the route with low delay information of the neighbor router

(ii) When both next-door neighbor routers of both routes are uncongested, select the route with low delay information of the neighbor router

(iii) When one next-door neighbor router is uncongested and another is congested, the route is selected on which next-door neighbor router is uncongested

Figure 5 shows that the flit is routed from route $P(1,1)$ to destination router $Q(4,4)$. The route from P to Q is shown through blue arrow lines, and R-D is the delay of the router.

4. Results and Discussion

For the experimental setup, a 4×4 mesh topology NoC having flexible routers is modeled. For routing purposes, the proposed routing algorithm is used in simulator tool

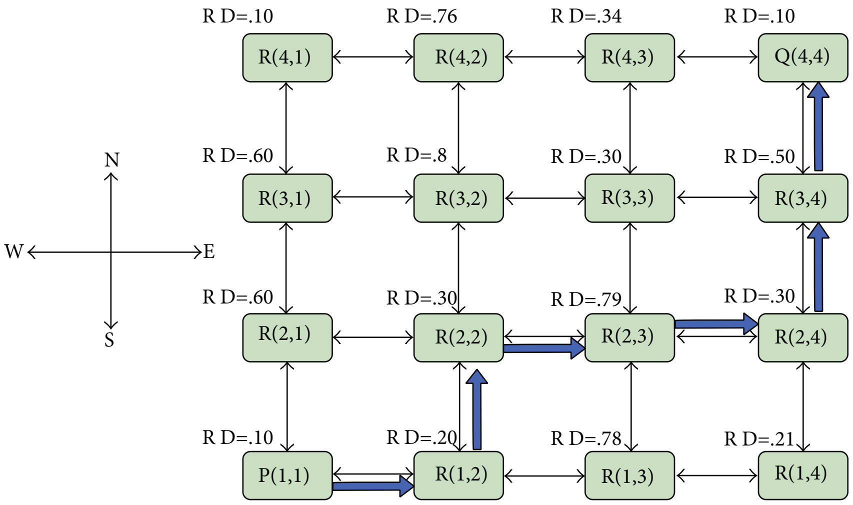

Figure 5: Routing from P to Q.

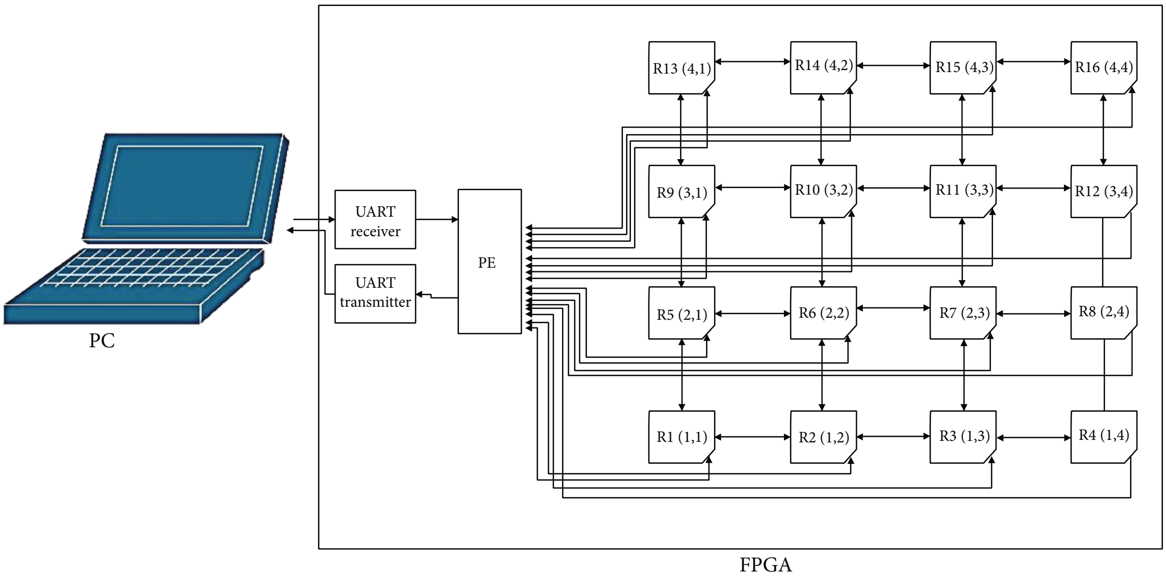

Figure 6: Experimental setup.

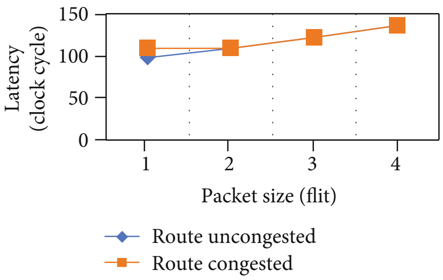

Figure 7: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}2}$.

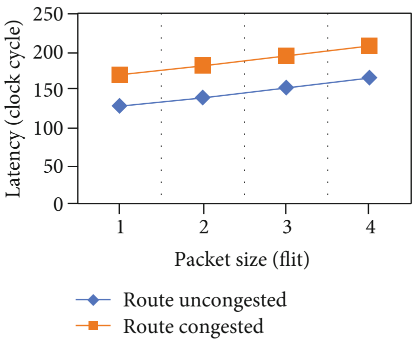

Figure 8: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}1}$.

ISE 14.7 webpack through Verilog language. Model NoC is implemented on the Spartan 6 XC6SLX9 FPGA development kit. UART is an intermediate node between NoC and PC for receiving and sending data from PC to FPGA kit. UART is also embedded/implemented in the FPGA kit and connected to NoC routers. Figure 6 shows the experimental setup and connectivity of PC, UART, PE, and 4 × 4 NoC.

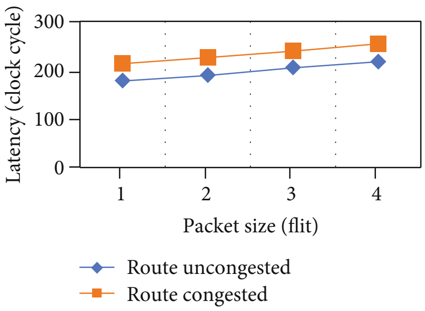

Figure 9: Latency from $R_{1\text{-}1}$ to $R_{4\text{-}1}$.

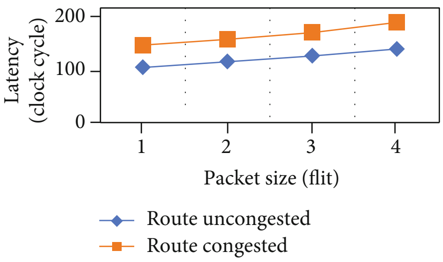

Figure 10: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}2}$.

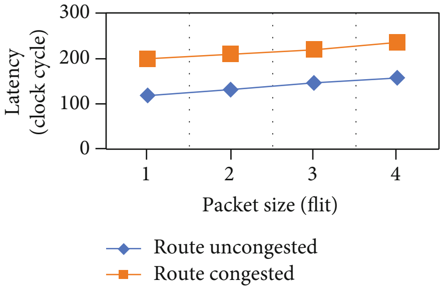

Figure 11: Latency from $R_{1\text{-}1}$ to $R_{2\text{-}2}$.

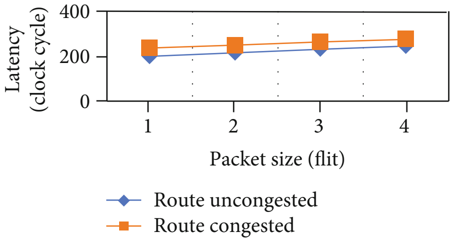

Figure 12: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}2}$.

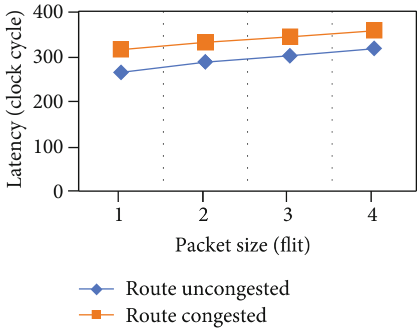

Figure 13: Latency from $R_{1\text{-}1}$ to $R_{4\text{-}2}$.

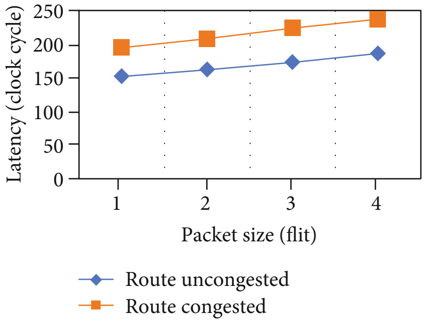

Figure 14: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}3}$.

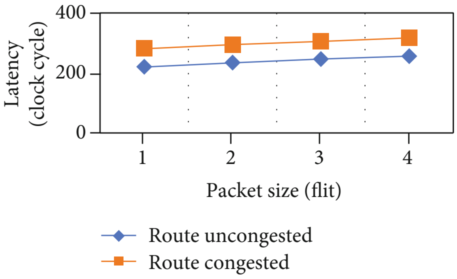

Figure 15: Latency from $R_{1\text{-}1}$ to $R_{2\text{-}3}$.

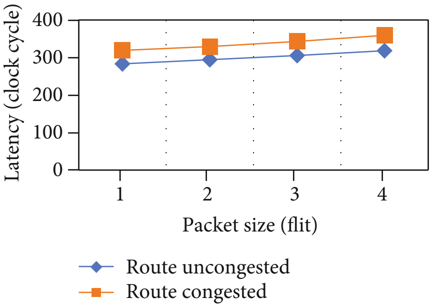

Figure 16: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}3}$.

For the result evaluation, an experiment was conducted by sending several packets having a size from 1 flit to 4 flits from source $R_{1\text{-}1}$ to all other destinations of NoC shown in Figure 5, and the latency was monitored in both situations, without congestion and with congestion. The flit size is 64 bits, and the size of the buffer is also 64 bits. One buffer in a router can store one flit at a time. The channel width is 8 bits, and in one clock cycle, 12.55 packets can be processed. The latency was analyzed in the worst case. The latency was explored through the Chip Scope Pro Analyzer. Average latency from $R_{1\text{-}1}$ to all other routers are shown in line graphs in Figures 7–21 and in bar graphs in Figures 22–36. In Figures 21–36, the blue line in the line graph and blue bar in the bar graph show the average latency from $R_{1\text{-}1}$ to all other routers when the route from $R_{1\text{-}1}$ to destination is uncongested. The red line and red bar show average latency

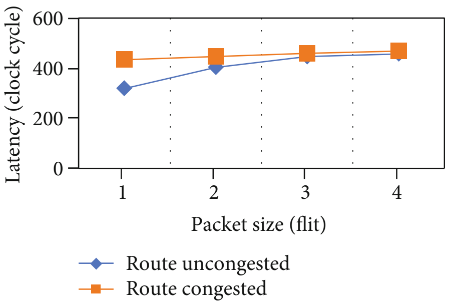

FIGURE 17: Latency from R_{1-1} to R_{4-3}.

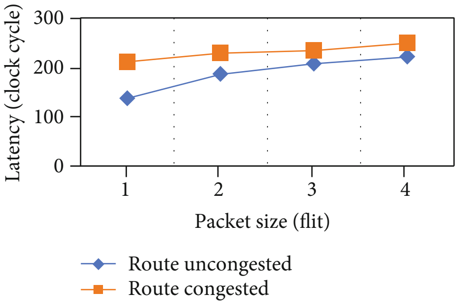

FIGURE 18: Latency from R_{1-1} to R_{1-4}.

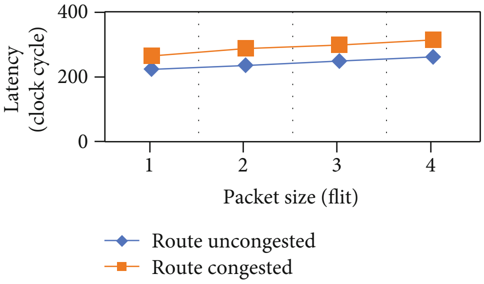

FIGURE 19: Latency from R_{1-1} to R_{2-4}.

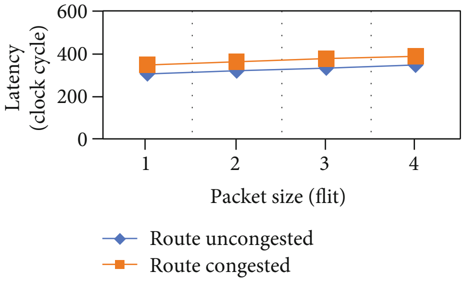

FIGURE 20: Latency from R_{1-1} to R_{3-4}.

when the route from R_{1-1} to destination is congested. The latency depends on the route length or the number of nodes between source and destination, congestion status of the route, and packet size.

The node between R_{1-1} to R_{1-2} and R_{2-1} is 0. The highest number of nodes in this mesh topology from R_{1-1} to R_{4-4} is 5

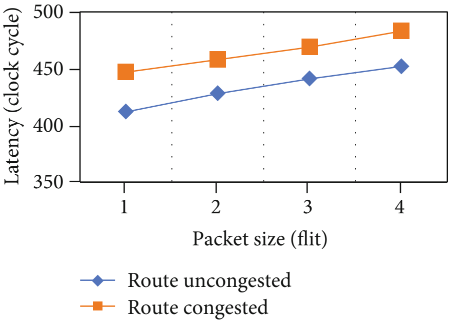

FIGURE 21: Latency from R_{1-1} to R_{4-4}.

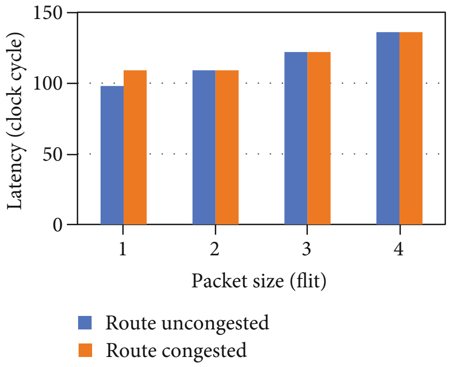

FIGURE 22: Latency from R_{1-1} to R_{2-1}.

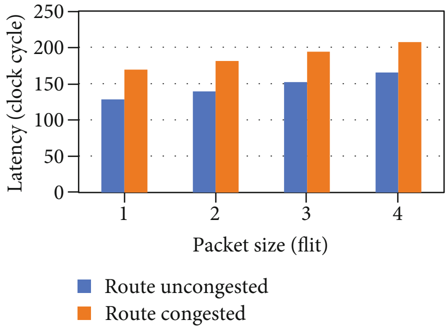

FIGURE 23: Latency from R_{1-1} to R_{3-1}.

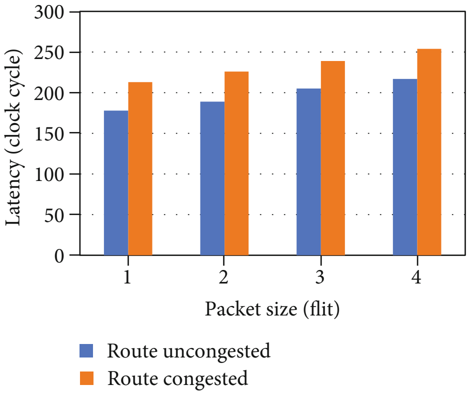

FIGURE 24: Latency from R_{1-1} to R_{4-1}.

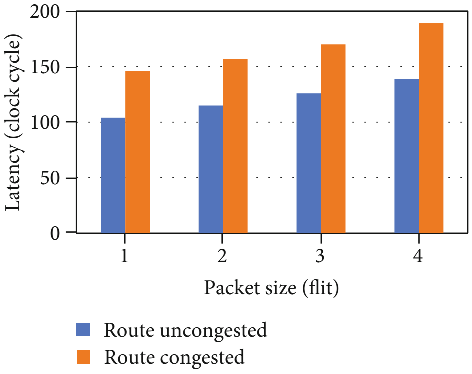

FIGURE 25: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}2}$.

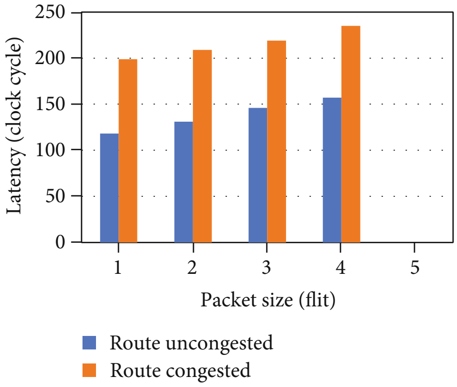

FIGURE 26: Latency from $R_{1\text{-}1}$ to $R_{2\text{-}2}$.

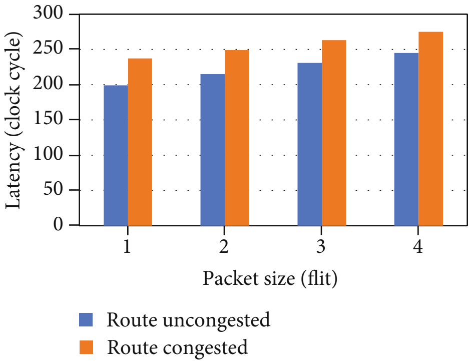

FIGURE 27: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}2}$.

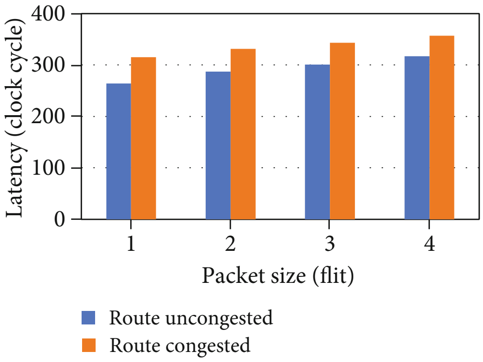

FIGURE 28: Latency from $R_{1\text{-}1}$ to $R_{4\text{-}2}$.

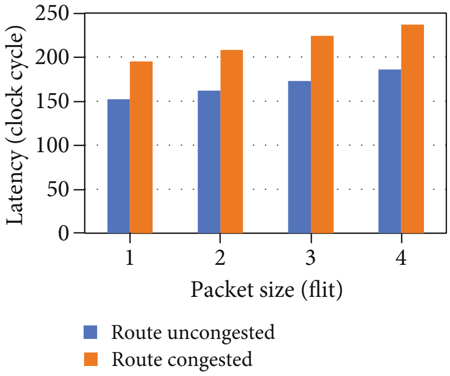

FIGURE 29: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}3}$.

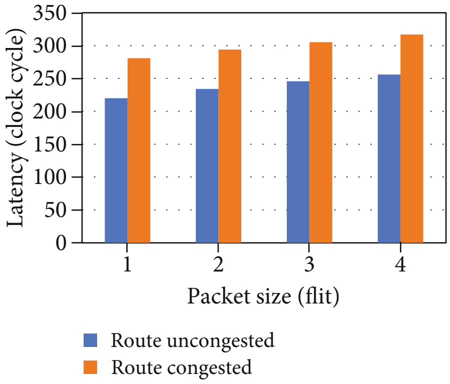

FIGURE 30: Latency from $R_{1\text{-}1}$ to $R_{2\text{-}3}$.

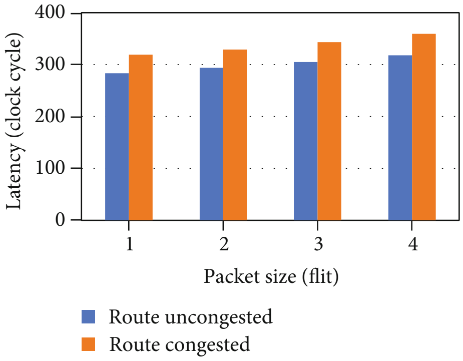

FIGURE 31: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}3}$.

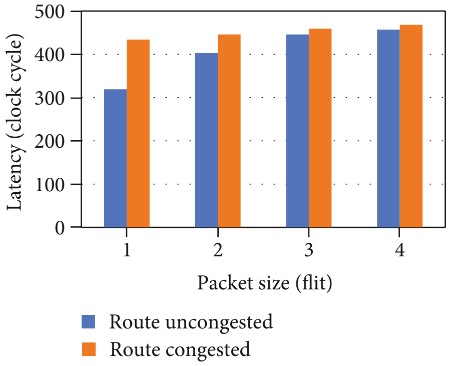

FIGURE 32: Latency from $R_{1\text{-}1}$ to $R_{4\text{-}3}$.

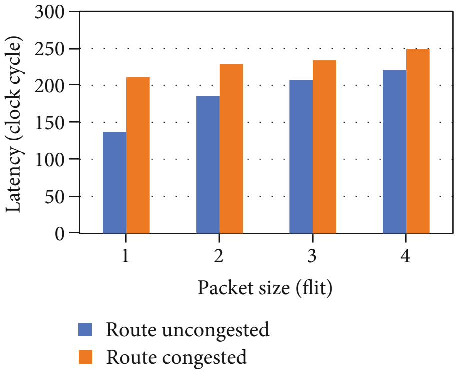

FIGURE 33: Latency from $R_{1\text{-}1}$ to $R_{1\text{-}4}$.

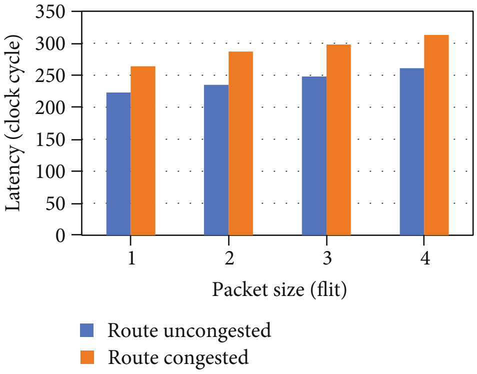

FIGURE 34: Latency from $R_{1\text{-}1}$ to $R_{2\text{-}4}$.

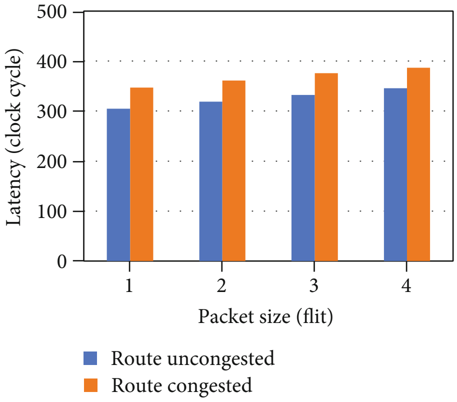

FIGURE 35: Latency from $R_{1\text{-}1}$ to $R_{3\text{-}4}$.

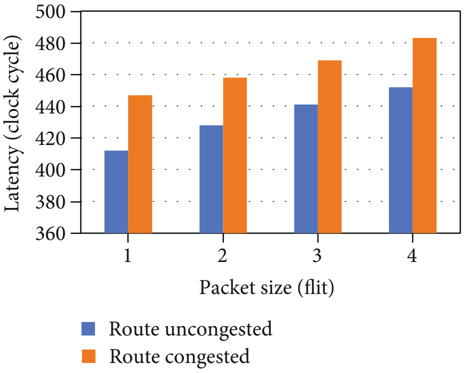

FIGURE 36: Latency from $R_{1\text{-}1}$ to $R_{4\text{-}4}$.

when a route is uncongested. The latency is increased when the number of nodes between source and destination increases. The latency of congested routes is higher than uncongested route latency because of computation in a router for finding an uncongested route from source to destination. In Figures 21–36, a congested route means finding an alternate uncongested route by the router based on shared congestion information when the first shortest route to a destination is congested.

In the experiment for the calculation of results, packet size is from 1 flit to 4 flits, and the size of flit is 64 bits. The latency in Figures 21–36 increases with the packet size.

5. Conclusion

In this paper, a congestion-aware routing algorithm is proposed in which the congestion information between routers is shared through a data flit. The routing decision in the router from source to destination is based on the congestion information of neighbor routers. The proposed routing algorithm is a congestion-aware adaptive routing algorithm. The routing algorithms in the literature use a separate network to share congestion information which increases the area of NoC. In few routing algorithms, congestion information is transmitted through the data line, and during congestion information sharing time, the data flits are not sent. In the proposed routing algorithm, the congestion information between routers is shared when the date flit traverses over the network. The proposed routing algorithm distributes the traffic over the network to avoid congestion in the router of NoC. The route from source to destination is dependent on basis of the congestion status of the NoC router. The packet/flit in the proposed routing algorithm is sent on an uncongested route or the route which has low congestion compared to other routes to a destination. From the experiment results, the average latency is calculated in both conditions, congested route and uncongested route in the worst case. The latency decreases up to approximately 50-70% when we increase the buffer size in router and channel width.

Data Availability

The data that support the findings of this study are available from the corresponding author, Ghulam Mohammad Karami (ghulam.karami@smec.com), upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] R. R. Tummala and V. K. Madisetti, "System on chip or system on package?," *IEEE Design & Test of Computers*, vol. 16, no. 2, pp. 48–56, 1999.

[2] T. A. Claasen, "An industry perspective on current and future state of the art in system-on-chip (SoC) technology," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1121–1137, 2006.

[3] W. J. Dally and B. Towles, "Route packets, not wires: on-chip inteconnection networks," in *Proceedings of the 38th annual design automation conference*, pp. 684–689, Las Vegas, NV, USA, 2001.

[4] S. Kumar, A. Jantsch, J. P. Soininen et al., "A network on chip architecture and design methodology," in *Proceedings IEEE computer society annual symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, pp. 117–124, Pittsburgh, PA, USA, 2002.

[5] G. Adamu, P. Chejara, and A. B. Garko, "Review of deterministic routing algorithm for network-on-chip," in *2nd international conference on science, technology and management*, pp. 741–745, India, 2015.

[6] M. A. J. Sethi, F. A. Hussin, and N. H. Hamid, "Review of network on chip architectures," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 10, pp. 4–29, 2017.

[7] A. V. de Mello, L. C. Ost, F. G. Moraes, and N. L. V. Calazans, *Evaluation of Routing Algorithms on Mesh Based NoCs*, PUCRS, Av. Ipiranga, 2004.

[8] A. Mejia, M. Palesi, J. Flich et al., "Region-based routing: a mechanism to support efficient routing algorithms in NoCs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 3, pp. 356–369, 2009.

[9] S. D. Chawade, M. A. Gaikwad, and R. M. Patrikar, "Review of XY routing algorithm for network-on-chip architecture," *International Journal of Computer Applications*, vol. 43, pp. 975–8887, 2012.

[10] A. Intel, *Touchstone Delta System Description*, Intel Corporation, 1991.

[11] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *ACM SIGARCH Computer Architecture News*, vol. 20, no. 2, pp. 278–287, 1992.

[12] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, "CATRA-congestion aware trapezoid-based routing algorithm for on-chip networks," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 320–325, Dresden, Germany, 2012.

[13] K. Ahmad and M. A. J. Sethi, "Review of network on chip routing algorithms," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 7, no. 22, p. 167793, 2020.

[14] S. Ganesan, V. Muthuswamy, G. Sannasi, and K. Arputharaj, "A comprehensive analysis of congestion control models in wireless sensor networks," in *Sensor Technology: Concepts, Methodologies, Tools, and Applications*, pp. 1194–1214, IGI Global, 2020.

[15] J. Hu and R. Marculescu, "DyAD: smart routing for networks-on-chip," in *Proceedings of the 41st annual Design Automation Conference*, pp. 260–263, New York, NY, USA, 2004.

[16] M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip," in *Proceedings of the 43rd annual Design Automation Conference*, pp. 849–852, New York, NY, USA, 2006.

[17] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "A heuristic path search for congestion control in WSN," in *Industry Interactive Innovations in Science, Engineering and Technology*, pp. 485–495, Springer, 2018.

[18] P. Lotfi-Kamran, A.-M. Rahmani, M. Daneshtalab, A. Afzali-Kusha, and Z. Navabi, "EDXY - a low cost congestion-aware routing algorithm for network-on-chips," *Journal of Systems Architecture*, vol. 56, no. 7, pp. 256–264, 2010.

[19] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "An improved congestion-aware routing mechanism in sensor networks using fuzzy rule sets," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 890–904, 2020.

[20] B. Wang, H. Cai, F. Qu, and Y. Yang, "The congestion controlling algorithm based on dynamic routing table on network on chips," in *10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2014)*, pp. 381–385, Beijing, China, 2014.

[21] C. Chen, Q. Li, N. Li, H. Liu, and Y. Dai, "Link-sharing: regional congestion aware routing in 2D NoC by propagating congestion information on idle links," in *2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM)*, pp. 291–297, Shanghai, China, 2018.

[22] M. Selvi, S. S. Kumar, S. Ganapathy, A. Ayyanar, H. K. Nehemiah, and A. Kannan, "An energy efficient clustered gravitational and fuzzy based routing algorithm in WSNs," *Wireless Personal Communications*, vol. 116, no. 1, pp. 61–90, 2021.

[23] R. Xie, J. Cai, X. Xin, and B. Yang, "MCAR: non-local adaptive network-on-chip routing with message propagation of congestion information," *Microprocessors and Microsystems*, vol. 49, pp. 117–126, 2017.

[24] S. Munuswamy, J. M. Saravanakumar, G. Sannasi, K. N. Harichandran, and K. Arputharaj, "Virtual force-based intelligent clustering for energy-efficient routing in mobile wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, pp. 1444–1452, 2018.

[25] S. T. Muhammad, M. Saad, A. A. El-Moursy, M. A. El-Moursy, and H. F. Hamed, "CFPA: congestion aware, fault tolerant and process variation aware adaptive routing algorithm for asynchronous networks-on-chip," *Journal of Parallel and Distributed Computing*, vol. 128, pp. 151–166, 2019.

[26] S. T. Muhammad, M. A. El-Moursy, A. A. El-Moursy, and H. F. Hamed, "Architecture level analysis for process variation in synchronous and asynchronous networks-on-chip," *Journal of Parallel and Distributed Computing*, vol. 102, pp. 175–185, 2017.