IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

Latency and Communication Reduction by Adopting the Better Flow Control Mechanism for Network on Chip

E G Satish[1], Ramachandra A C[2]

[1] Research Scholar, Department of Computer Science and Engineering,

Nitte Meenakshi Institute of Technology, Bangalore, India and affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India,

[2] Professor & Head Department of Electronics and Communication Engineering,

Nitte Meenakshi Institute of Technology, Bangalore, India.

Abstract: Today, a potential alternative known as network-on-chip (NoC) is being used in multicore systems to circumvent the limitations of traditional on-chip networks. NoC designs enable the use of high-capacity wireless connections to significantly decrease the latency associated with multi-hop communications. This article discusses a new technique for increasing the performance and power usage of NoCs. The suggested method makes use of a customized version of the RED algorithm's queue length concerns. A constantly increasing set of cores on the device need a scalable architecture; this is the most essential option. Mesh-based NoCs are the most commonly available topologies in many-core processors today as a scalable alternative to the conventional shared bus. To ensure long-term scalability and performance, the low-latency connection between the cores becomes more important. The delay between endpoints in an ideal network is roughly equal to that of a single cycle. The suggested method was tested using a variety of simulated traffic patterns and the SPLASH-2 trace-driven benchmark suite. The testing findings show that the algorithm significantly lowers latency and power usage when compared to a traditional NoC.

Keywords: Routing algorithm, Power efficiency, Latency, Topology, Interconnection Network, Network-on-chip

1. INTRODUCTION

The frequency of cores on circuits is rising as chip manufacturing technology advances. As a result of its scalability and performance, networks on the chips have become the primary communication theory for multi-core processors. While the size of NoC is increasing, researchers are searching for ways to optimize characteristics like latency, power usage, and area utilization. Generally, the characteristics of NoC may be classified into two categories: performance and cost. Throughput and average packet delay are performance metrics, whereas power consumption, area overhead, and peak temperature are cost metrics. Numerous approaches for optimizing the characteristics of networks-on-chip are discussed, including application map-based approaches, architecture-level solutions, routing protocols, router design techniques, and pour direct methods [1,2]. All these techniques are designed to get better the cost and presentation of the network. This article offers a method for decreasing communication latency, which is a critical issue in NYC. A packet must be routed via intermediary routers and network connections. Thus, in networks-on-chip, the gateways and communication connections are the most critical delay factors. The following equation is used to determine the latency of a packet depending on the clock cycle (1)

$$T = H.t_r + H.t_w + T_c + \frac{L}{b} \qquad (1)$$

Where H denotes the number of hops, tr denotes the router's delay, tw denotes the delay of the interaction wires between two switches, Tc denotes the router's congestion delay, and L/b denotes the serialization delay for both the body and tail flits. Packet latency is improved by one cycle when using intermediate node latching. As a consequence of each router's flit latching, there is an increase in H in Equation 1. The number of hosts (H) and the packet latency grows proportionally as the network's dimension is raised. It is recommended that no intermediate nodes should be used since doing so greatly reduces the latency caused by these nodes. This figure illustrates how flits traverse a network path without having to latch onto the wire between the source and destination [3]. The optimal network latency in this instance is equal to T = 1 + L/b.

IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

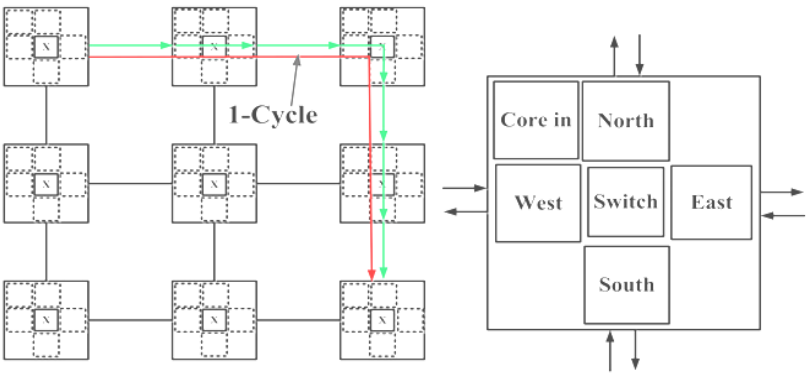


Figure.1. Single-Cycle Traversal and Baseline Router

Numerous improvements in the field of NoC architecture have been suggest indirectly to decrease power utilization and enhance recital. They have been suggested to address a variety of on-chip network design challenges, including layout design, router microarchitecture intends, mapping techniques, and shifting and pour regulation apparatus design[4,5]. allowing for the improvements stated before, pour management and buffering problems are critical for power saving. In NoCs, buffer management and buffer size are critical factors. The size of each NoC router's input channel buffers has a considerable effect on the overall chip area. Raising the buffer size for each input signal from two to three words, for example, improves the router size by 30% in a 4x4 Mesh NoC. Meanwhile, depending on the network traffic, increasing the buffer size may delay the network's saturation phase transition. In most other words, buffer maintenance is intrinsically linked to the network's flow control rules. In turn, flow control affects network efficiency and resource use. Thus, an effective flow control strategy may boost the system performance to up to 80% of its calculated values, while a poor policy may achieve just 30% of such a value. In light of the above, the dynamic buffer allocation technique has been suggested. In this approach, buffers are created for each router on a per-router basis and then deallocated [6,7].

The next section will examine some of these methods. Congestion is a significant problem in interconnection networks. Indeed, congestion is often seen as critical to achieving high performance and efficiency. When bandwidth demand exceeds available capacity, network congestion develops. Congestion leads to increased latency and decreased efficiency in lossless networks. As a result, the absence of an appropriate congestion management strategy results in a substantial decrease in the entire connection speeds or a subset of it. When a VC transmission is stopped, subsequent packets may use another VC to traverse the physical connection. As a result, VCs enhance the functioning of physical channels and boost network performance overall. In Figure 2, a physical channel and its associated VCs are shown [8]. Technically, VCs may be used to any traffic method of control to decrease the likelihood of headers being blocked. The random Early Detection method was originally developed to address the congestion management issue in TCP-based networks. Before packets join the queue, the algorithm admits or rejects them depending on the likelihood of queue fullness. This method avoids the queue from being overflowing prematurely and achieves a compromise among applications with such a high rate of input and those with a low rate of input. The method delays the queue's filling.

The simplest method to avoid congestion is to discard messages when the caches are full. This technique, which is also the basis for the RED, is acceptable on lossy networks like the Internet. The delay associated with re-sending packets is often acceptable on these networks. On the other hand, the system cannot often tolerate this delay with NoCs and High - speed Computing (HPC). As a result, other methods must be used in such systems [9,10]. The method presented in this article makes use of the RED algorithm's queue length concerns to optimize the efficiency and energy consumption of NoCs. Additionally, a stochastic learning-automata-based method was used to improve the RED algorithm's needed threshold values. The RED method has been adapted for use with NoCs in this approach. Given that the network cannot tolerate packet discarding, we forego packet jettisoning in NoCs and instead tackle the threshold values. Additionally, a novel router microarchitecture is shown in which variable buffer size is used to regulate the flow of virtual circuits. Along with increasing buffer space usage, this change may also improve network performance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

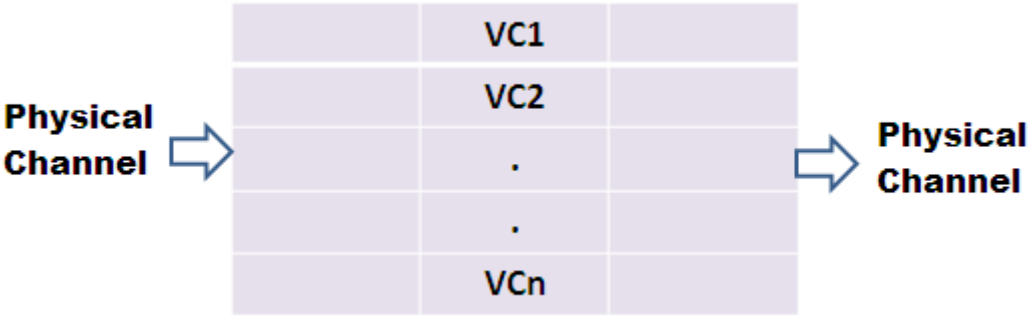


Figure.2. Virtual and Physical channels Match.

To allow inter-process communication, which may be done with certain restrictions and characteristics in mind, we need on-chip communication. When it comes to seeing network-based process-to-process communication, one may see the OSI model in action as stated in the ISO's OSI reference model. A model consisting of seven distinct layers was devised to link diverse systems, which are heterogeneous. and have been disseminated [11,12]. In general, the layered structure breaks down communication. dividing a complex issue into smaller, simpler parts at several hierarchical levels Rather In contrast to a monolithic framework, many layers are implemented, each of which tackles one specific problem. A significant part of the issue. What I'm trying to say is that layered design offers a more modular design. Each layer of a protocol or service specification is independent of the other layers. Layers belong to the same layer, which means peer entities from the same layer may interact with one other transparently. If the services are being added to a single layer, then just the functions would need to be modified. Making use of the functionalities that are already available at all other levels. as a result of With this approach, many NoC groups have used these benefits. We have developed an adaptation of it for use in building a protocol stack for on-chip communication [13].

The platform, NoC, will provide well-defined application programming interfaces. as well as IP integration There are two layers of interfaces that may be distinguished. A hardware core interface that connects the cores is called a Core-Level Interface (CLI). It's at this stage that the interfaces such as AXI, OCP, and VCI are used when it comes to IPs and CPUs. Such core applications as Core Connect and DTL. On the other level, hardware integration is part of the interface. With the help of a connection adapter, connect logic and embedded software development. OS (sometimes known as an operating system) and middleware components are often part of the overall platform. This Application-level interface level (ALI). a recent suggestion has been put forth Multiprocessor on-chip interfaces with two-level implementations may be found in [14].

An on-chip communication model incorporates both the horizontal and vertical perspectives of hierarchical interface-based communication and interpersonal communication although it has been an open question Consider Figure.3. Separately, the two viewpoints are consistent, as shown in the figure. Be seen, using examples like P1, P2, P3, and P4 shown in Figure the ALI, which is often used for application-layer purposes, is being utilized here. The connection between the hardware and software merging hardware cores and middleware session and transit layers are realized in software cores, which are connected to the CLI. The CLI provides a mechanism for packaging a network [5,16]. when skipping a layer is worthwhile As long as the interfaces match, it is feasible. Consider the case of a hardware IP that implements the CLI; instead of connecting to the ALI, it may be directly linked to the CLI.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

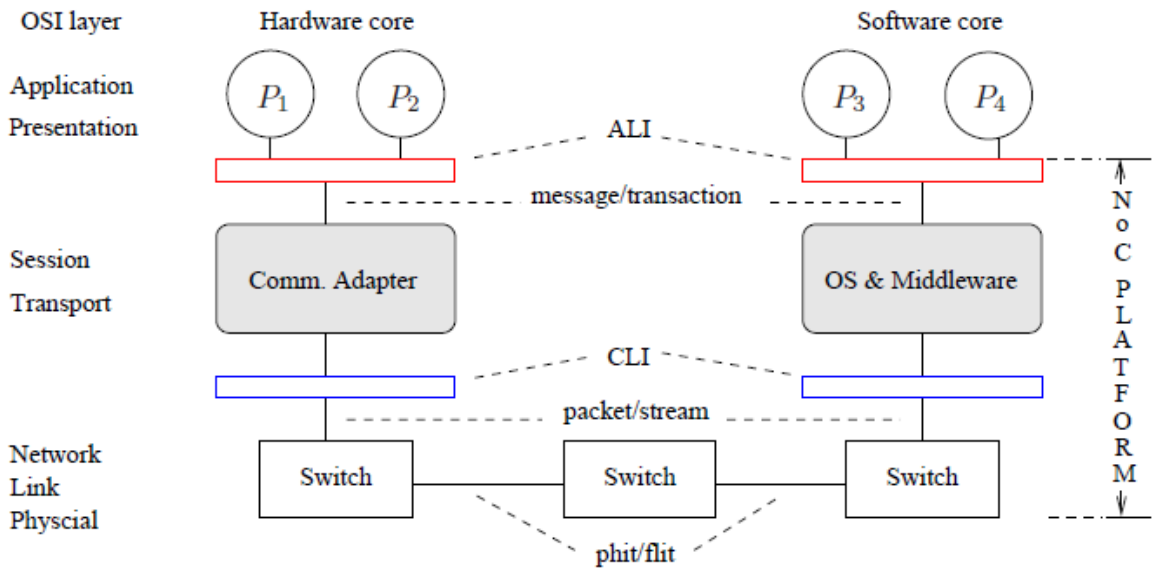


Figure.3. Application-level interconnect with Core-level interconnect

2. RELATED WORK

Our lives have been significantly influenced by the exciting advances in contemporary electronic technology, pervasive and ubiquitous computing, ambient intelligence, communication, and the Internet are sometimes referred to as ubiquitous computing. Today, microelectronic devices have a significant impact. The modes of communication, education, and recreation. The primary motivator The System-on-Chip (SoC) technologies are responsible for decades of advancements. where several ULSI chips are used to combine complicated applications [17]. Not just on a functional level These goods, which include mobile phones, notebook computers, and personal computers, have been enhanced. Handheld sets are getting quicker, more compact, more capable, and less in heaviness. One might reasonably believe that. This tendency will continue indefinitely. We may incorporate more if this trend continues and even systems onto a single chip. Nonetheless, Because existing methods for SoC design and integration are not uniform in their advancement, the enormous difficulties that have been faced. As circuits operate at increasing frequencies, minimizing power consumption becomes critical. Power is a design restriction that takes precedence over performance [19]. Despite advancements in process and circuit design, power consumption continues to increase at a fast rate. Equally worrisome is the linear rise in power density on the chip die. Reduced electricity usage becomes more difficult in the face of DSM impacts. Indeed, a 10% to 15% decrease may result in a twofold increase in leakage current. Leakage will become the primary cause of power consumption in ever-smaller electronics. Additionally, leakage occurs when electricity is flowing via the circuit. As mentioned in the ITRS roadmap, the biggest danger to the semiconductor roadmap's continuance is the expense of design.

This article [22] investigates flit ejection models in the context of a wormhole virtual channel transition. Rather than the expensive ideal flit-ejection model, two alternatives are presented that significantly decrease the buffering cost. Experiments demonstrate that when the network is not overloaded, the p-sink model provides almost identical performance to the ideal sink model. The author defined the flit-ejection issue, suggested solutions, carried out tests, and wrote the paper. This article [20] discusses a connection-oriented multicast method for wormhole switched nodes of contention. A multicast process in this system consists of three phases: setup, communication, and release. Our studies demonstrate that although multicast increases throughput, it has no discernible effect on unicast performance on a network with a mix of unicast and multicast traffic. The author contributed to the conception and design of the protocol, proposed experimental techniques, and drafted the article. TDM and VCs on network-on-chip must be conflict-free in addition to being allotted appropriate time slots. We propose and verify theorems that provide sufficient and necessary criteria for the establishment of conflict-free VCs using the extended notion of logical networks. Additionally, we formulate the multi-node VC configuration issue and provide a backtracking method for constructively exploring the solution space for solutions. The author conceived the problem, established and proved the theorems, created the software, performed experiments, and produced the paper. This article [21] discusses the traffic configuration techniques that have been created for NNSE. It introduces a single language for configuring uniform and localized traffic, as well as an application-oriented traffic setup for on-chip network assessment. The author developed application-oriented traffic and established the unified expression for regular traffic patterns, integrated the techniques into NNSE, performed tests, and authored the paper. These results may be utilized as recommendations by designers to help them make the best architectural choices

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

for the deflection network [22]. The author defined the issue, develop possible solutions, and produced the text. The article offers a technique for examining the feasibility of transmitting mixed real-time and non-real-time communications in wormhole-switched networks. It presents a contention tree model for estimating the worst-case performance of real-time message delivery. The author created the contention tree model, the algorithm, the software, conducted the tests, and authored the paper.

3. PROPOSED MODEL

A widely accepted notion in the realm of telephony, computer communication, and parallel machines is the communication network. On-chip networks are similar to these networks in many ways but are distinct as well. Clear communication requires that throughout the article, we use the term micro-network to refer to on-chip networks, parallel-machine networks are called macro-networks, and telephone or computer networks are referred to as tele networks. On-chip networks are created and built for closed systems that are dedicated to various applications such as heterogeneous ones. Designed on distributed boards, parallel-machine networks are created. Applying certain algorithms in a given way is common for that application. The use of computers in networks is spread throughout the globe, and they're intended for open systems. Using client-server, peer-to-peer, and multicast applications flexibly. Telephony networks are similarly spread across geographic regions but are configured primarily to provide communication channels such as audio, video, and data. the appearance of a ensures customization in which the network features, including its physical characteristics, may be adjusted to meet individual network needs the information which relates to the network-level, link-level, and physical characteristics may be sent to others. The combination of communication and processing at the application level may be effectively optimized. Wide-parallelism is possible on a micro-network since the whole network is constructed on a single chip. It enables synchronous high-rate clocking with the use of wires. Conversely, it possesses a lot of The performance, area, and power limits that are inherent to tradeoffs on how to develop an SoC. Data transmission is communicated via conversation. a rank-and-file member Delay is the toughest criterion for on-chip networks. And there's also jitter. A nanosecond is one-billionth of a second. the vast majority of potential applicants. Many of the software-based advanced arbitration, routing and flow-control algorithms are based on artificial intelligence algorithms. most SoCs have cost as a significant constraint for on-chip networks large amount of sales. Most buffering in on-chip networks is found in a relatively small area As well as board-level, local-area, and wide-area networks, it is costly in comparison. The key aspect is that it limits the number of routing tables and virtual channels that are permitted on a NOC. Network nodes have buffers. All types of electrical systems, no matter how little, need energy consumption [23]. networks. Additionally, integrated circuit networks are created for embedded applications as well. depending on battery-powered gadgets. More processing power is required for these kinds of applications. On a larger scale, it can't be compared to large-scale networks. On-chip, we also stated DSM effects challenge network architectures. conquering adverse physical side effects to have the same level of importance as network design itself. Also, various SoC network designs are created. For various use cases, it functions not just as a single solution. Therefore Much attention should be paid to reconfigurability in network architecture. Network-on-Chip is a revolutionary advance, not an evolutionary one. out-of-the-box thinking approach to the SoC design dilemma our attention moves from computation to communication. Early attention should be given to connection in the design. A platform-based design approach allows processes to be set while favoring meet-in-the-middle results. The tendency is toward a top-down or bottom-up methodology. These characteristics of NoC are:

As technology grows, the scope of what can be done in one cycle of the clock becomes smaller. Due to this, chip design is rapidly becoming more about communicating with other devices, rather than having more computing power. To make use of the large available chip capacity, all of the chips must be divided into several areas. To aid in the management of the characteristics of long wires, which include the intermediate layer and the top layer, excellent partitioning should be regular. Each module is partitioned into regions, and each partition is independent. Using global synchrony and global wires may reduce dependence on global synchrony. Registers may be utilized in wire segments to ensure that the overall latency of the design is insensitive to register operation [24]. That is, an IP may be connected to a switch, rather than a single network interface card (NIC). Networks use switches that are linked to each other to route packets. The link and physical layer signal and power integrity problems may be handled, as well as the upper layer ones. The physical dimension may be made redundant via the use of temporal, spatial, and informational redundancies in the transmission process to improve system dependability. Design robustness and dependability may be enhanced by organizing the communication and minimizing the influence of the DSM symptoms. Communicating with IP modules partitioned into individual chips allows the creation of a naturally parallel network. It is possible to implement more cores if the chip capacity is not exceeded. Total network capacity is shared across the inter-core communications, with a very high degree of concurrency. To meet the bandwidth requirements of the application of interest, the network may be dimensioned to fit the needs.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

Concurrent processing in computation and communication is possible due to the parallel architecture. This boosts performance, reduces power use, and is sequentially scalable. A protocol stack is usually designed to hide the communication that occurs over the network. Layers provide specific functions, use well-defined protocols, and provide specified interfaces [19]. For each layer, the design space needs to be thoroughly examined. Designers, Analysts, and implementers must take into consideration the compromises between performance and cost while designing, analyzing, and implementing the communication architecture. System-wide performance analysis and service quality are critical in addressing unpredictability. Given the huge hurdle of the design cost, a programmable, reconfigurable, and extendable communication platform is needed for creative and sophisticated SoC designs.

The goals are as follows: To provide a hardware communication architecture, an accompanying interconnect, and together with low-level protocols to provide integration with custom circuitry and for software programming. Reuse of the architectural level is enabled by this. An issue to overcome is determining the proper optimization. A platform must support more than one application, but it must also meet the requirements of numerous apps that belong to the same domain. Furthermore, designers will need to customize designs to increase their performance and efficiency to gain an advantage. Enabling interfaces and functional blocks to be reused, layers is critical, as it supports IPs and functional blocks. Interface standardization is a significant issue since IPs from various manufacturers are not exchangeable unless they adhere to the same standards. It should be efficient, but it should also include older IP addresses. IP reuse and integration should be maximized in the design of a NOC. When component and architectural validation are used in a design flow, verification time, market-time, and quality assurance may all be reduced, increasing design productivity.

The nodes communicate with one another by creating data packets and transmitting them across this communication infrastructure. A crossbar of dimension PxP connects the digital outputs of a router having P intake and P output ports. Thus, NoC-focused research does not just cover multiple SoC design elements but instead has the potential to transform into a new field [22]The design process and its related toolchain are included. RED of packets is a technique used in active queue management. In an internet network, the algorithm is regarded as a congestion management technique. In comparison, NoCs have been subjected to the Drop tail algorithm. When the buffer is complete and the network has reached saturation, this method discards packets. In comparison to drop tail, the RED algorithm more evenly distributes buffering intervals between traffic flows. RED's Classic method determines the total queue length and rejects packets depending on the statistical probability computed. No packets are destroyed in the method we employed, guaranteeing that the threshold is not exceeded due to package skipping. Entering this area is utilized solely to dynamically assign additional buffer to the queue in our method [30]. The parameters and notations used in the explanation of the RED algorithm are given in Table 1. We may compute the mean rating of the queue length by applying a low pass filter on the current queue length and the average arrival time, indicated by the variables Wq and Avg, respectively. This is accomplished by

$$\mathrm{Avg} = \left(1 - \mathrm{W_q}\right)\mathrm{avg} + \mathrm{W_q}\ \mathrm{queue_{len}}$$

Where represents the queue's length, whereas the average parameter indicates the queue's current length.

The method is based on an average packet delay and two model parameters, one for the lowest (min th) and one for the maximum (max th). packets are dropped with a specified probability., P_a, which is expressed as

$$\mathrm{P_a} = \frac{\mathrm{P_b}}{1 - \mathrm{P_b Count}} \qquad (2)$$

wherein the count is the data packets received before the previous drop and Pb denotes the probability of instantly marking [18], which is computed as

$$\mathrm{P_b} = \max_{\mathrm{p}} \left(\frac{\mathrm{avg} - \mathrm{min_{th}}}{\mathrm{max_{th}} - \mathrm{min_{th}}}\right) \qquad (3)$$

max is the greatest probability of a packet being dropped. As a result, the quantity of Pa (from Eq. (2)) will rise, resulting in an increment in the number of wasted packets. Fig.5 illustrates the quasi and the graph of the queue length decreasing, which is most likely proportional to the current queue length. Numerous research has been conducted to limit the wait length to an administrator-specified value. This method has a basic flaw: it lacks the adaptability to changing traffic loads, resulting in a lack of sustainability. Other approaches assume the threshold value to be constant.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

Initialization:

```
1. Avg=0
2. Count=-1
3. For each packet that arrives
4. begin
5. calculate the new average queue size avg:
6. if the queue is nonempty
7. begin
8. Avg=(1-Wg)*avg+Wq*queue_len
9. end
10. else
11. begin
12. m=f (current_time-empty-queue_time)
13. Avg=(1-Wg)*m*avg
14. end
15. if minth<=Avg < maxth
16. begin
17. Increment count
18. Calculate the probability Pa:
19. Pb=maxp (Avg-minth)/(maxth-minth)
20. Pa=Pb/ (1-count *Pb)
21. Drop the arriving packet with probability Pa and then count=0
22. End
23. Else if (mxth<=Avg)
24. Begin
25. Drop the arriving packet
26. Count=0
27. End
28. Else count =-1
29. End
30. When queue becomes empty
31. Empty_queue_time = current_time
```

(a) The pseudo-code

4. ROUTING ALGORITHM

The routing algorithm selects the network routes for the packets to follow. More often than not, the set of routes that are open to consideration is a smaller group of viable paths. There are three types of routing algorithms: deterministic routing, oblivious routing, and adaptive routing. A deterministic routing mechanism will select the same route every time, even if the source and destination nodes are different. Ignores the route variety of the network and does not account for the network's current condition. Due to the lack of special requirements, the load imbalances that may occur in the network are very easy and cheap to implement. In addition, providing the ordering of packets is easy to do this manner. Deterministic algorithms are seen as a subset of oblivious routing, which also contains random algorithms that evenly distribute traffic. Unmindful algorithms, on the other hand, do not consider the current network topology while generating routing choices. In reaction to the network status, adaptive routing distributes traffic dynamically. the condition of a node or connection or the length of queues may be included in the network state and load network-generated history. In other words, every hop must decrease the distance to the goal. If you aren't minimal, then you are non-minimal. Algorithmic and table-based routing mechanisms may be employed to accomplish the same goal. Routing tables are used at either the

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

source or each hop along the route, depending on the table-based routing method being used. Algorithmic routing doesn't save the route relation in a table. It is implemented most often as a combinational logic circuit for the sake of speed. Table-based routing tends to limit the algorithmic routing to simple routing algorithms and basic topologies, at the expense of generality. provides details on past network load. minimalist routing if every hop must decrease the distance to the destination, i.e., if it only routes packets along the shortest pathways to their destinations, it's known as shortest path routing. If it is not minimal, it is non-minimal. Algorithmic and table-based methods may be employed to implement routing. In this method, the routing tables are located at either the source or at each of the destinations along the route. Algorithmic routing instead of using a database to store the routing relation. It is implemented most often as a combinational logic circuit for the sake of speed. Table-based routing tends to limit the algorithmic routing to simple routing algorithms and basic topologies, at the expense of generality. Deterministic or oblivious minimum routing tends to result in simple switch designs, as compared to adaptive routing because routing decisions are made regardless of the network's current state.

Because packets must constantly execute on a cycle-by-cycle basis, packets are not packet queues. When it comes to prioritizing traffic, a deflection policy prioritizes packets on the usage of preferred connections. Packets are sent through the shortest possible routes when there is no congestion. Higher-priority packets prevail in arbitration and utilize preferred connections while lower-priority packets are sent to inferior routes. In optical networks, buffering optical data is prohibitively costly. In simple words, the combination of simplicity and adaptability make it popular for communication networks such as the Connection Machine. In addition, because of the reasons outlined above, on-chip networks in the Nostrum NoC have also been suggested. By using deflection routing, we can cut switch designs to half size. Designing a routing algorithm to minimize deadlock and livelock is the main issue while developing it to guarantee the proper functioning of the network. According to it is possible to prevent stalemates by incorporating application knowledge. Maximum delivery limits for deflection networks are found in [8]. The networks, thus, are free of livelock. Method 1 illustrates the pseudocode for the bypassing function, whereas Algorithm 2 illustrates the suggested routing algorithm in action.

The proposed routing method is similar to adaptive routing in that glides are propagated down the bypass channel and sent without latching via the router. In this instance, the routing algorithm mechanism does not route flits.

```
1.  Input: Incoming flit Output: Bypass flit
2.  for each direction of router do
3.  if incoming flit is header flit then
4.  Route incoming flit;
5.  if incoming flit is bypass flit then
6.  Send flit to the output;
7.  The output port is unreachable;
8.  end
9.  else
10. Buffer the incoming flit;
11. end
12. end
13. if incoming flit is body flit then
14. if incoming flit is bypass flit then
15. Send flit;
16. end
17. else
18. Route and buffer the flit;
19. end
20. end
21. if incoming flit is tail flit then
22. if incoming flit is bypass flit then
23. Send flit and release the output port;
24. end
25. else
26. Route and buffer the flit;
27. end
28. end
29. end
```

Algorithm 1: Bypass algorithm

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

Input: Inflow flit as output from the output port
Inflow flit (Routing)
If inflow flit is bypass flit then
Route it without the wait
End
If inflow flit is header flit then
Route as state as adaptive routing
If communicated with output port through the bypass then
Make available to bypass flit for output port by freeing
End
Else
Reserve the output;
End
End
Else
Route incoming flit according to the stated adaptive routing algorithm
End
If inflow flit is tail flit then
Free the output port
End

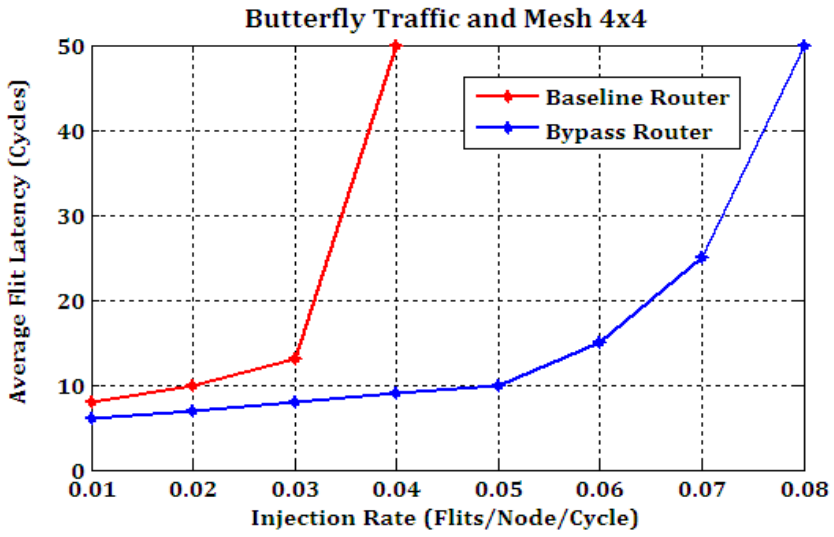
Algorithm 2: Algorithm for Semi-Adaptive Routing

5. SIMULATION AND RESULTS

The parameters for setup are shown in Table 1. Utilizing a range of traffic patterns and the XY routing algorithm, we begin by running just one bypass router in 4x4, 8x8, and 16x16 mesh topologies using a single bypass router. The latency of a one-dimensional bypassing router is shown in Figure.6. in comparison to the latency of a baseline router. The router simulations showed here demonstrate a reduction in the average packet latency on a per-connection basis. This is because flits are not locked in each router, resulting in a reduction in the number of hops (H) in Eq. 1.

Table.1 Target system and configuration

Technology	45nm	Topology	4x4, 8x8, 16x16 Mesh
Vdd	1.0	Router ports	5
Frequency	1.0GHz	Routing	XY, Proposed routing
Link Length	1mm	Flit width	128 bit



(a) Butterfly Traffic and Mesh 4x4

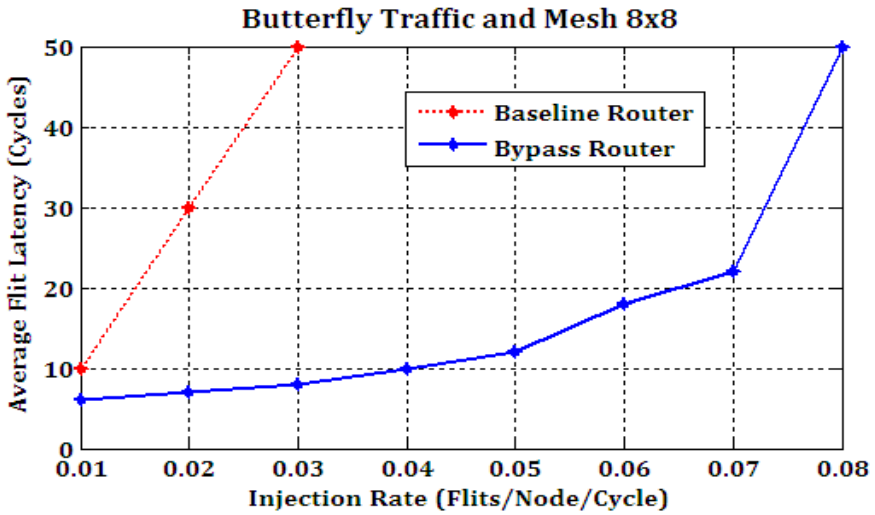
IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

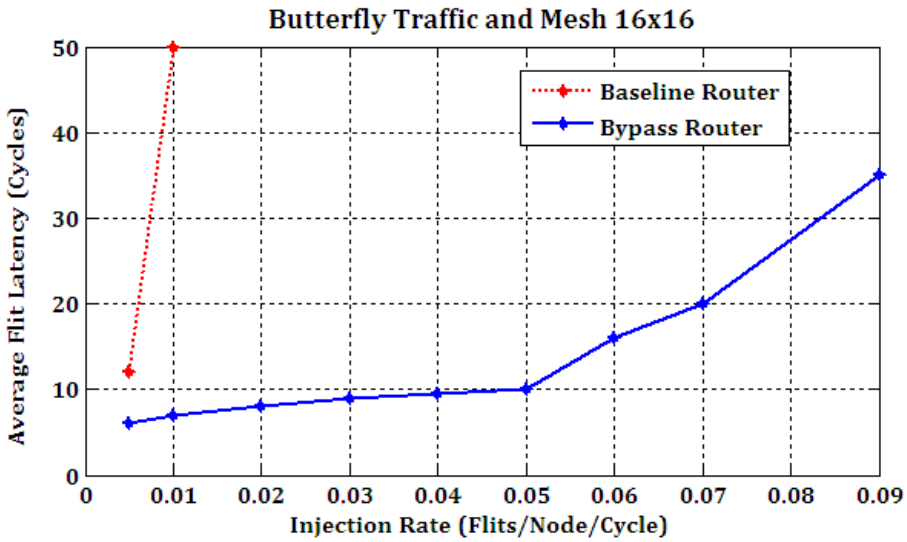
International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

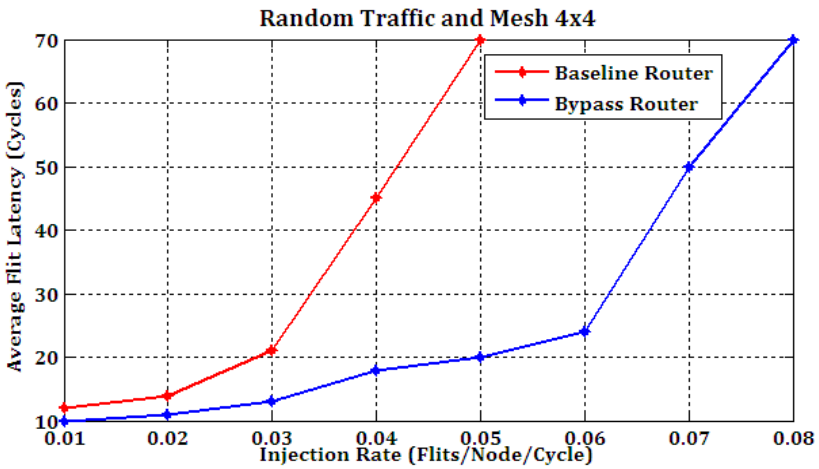
DOI: 10.17148/IJARCCE.2021.101239



(b) Butterfly Traffic and Mesh 8x8



(c) Butterfly Traffic and Mesh 16x16



(d) Random Traffic and Mesh 4x4

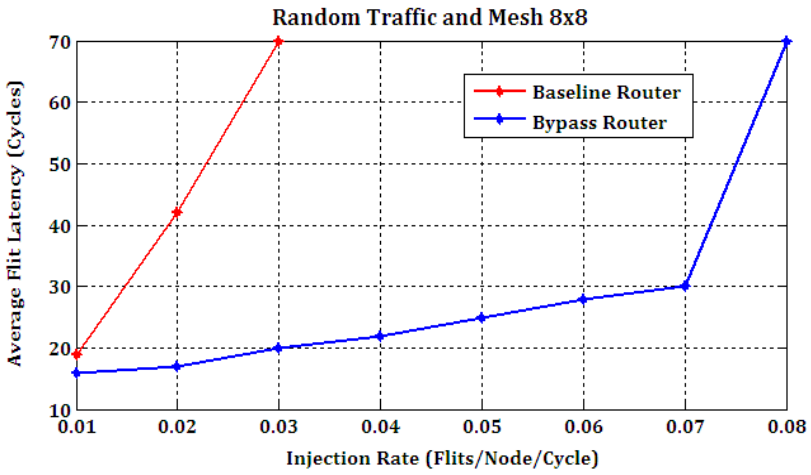
IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

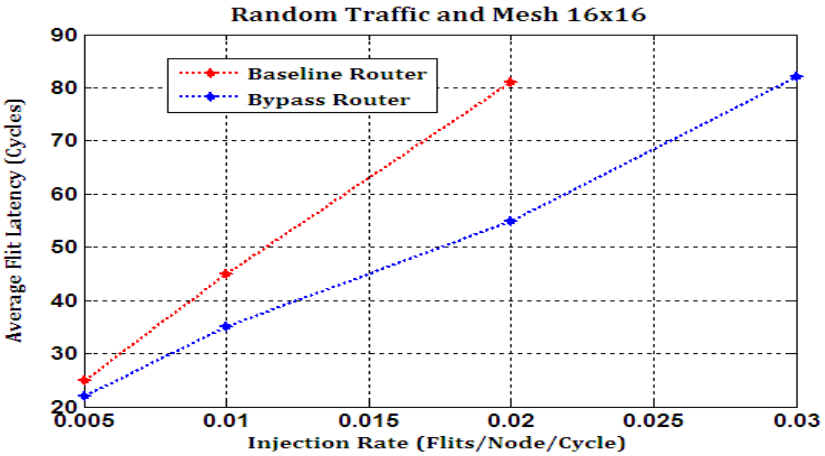
International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239



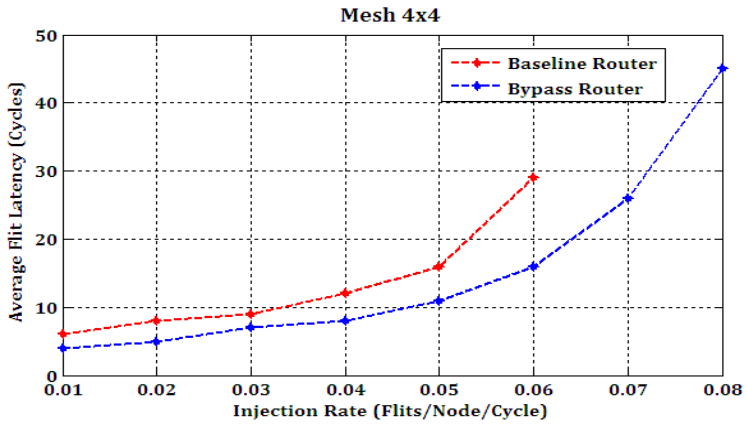
(e) Random Traffic and Mesh 8x8



(f) Random Traffic and Mesh 16x16

Figure.6. Bypass router with different synthetic traffic and XY routing algorithm

The suggested routing protocol and Butterfly traffic dispersion are used to mimic a one-dimensional bypass router in a 4x4, 8x8, 16x16 mesh. In Fig.7., the simulation results are compared to those of a baseline router operating in the same architecture and using the same adaptive routing method. As shown, deterministic routing and suggested routing algorithms significantly decrease latency in a one-dimensional bypass router. Bypass routes enable packets to transit intermediate routers without latching, decreasing the number of hops separating sender and receiver and also lowering average packet delay.



(a) Mesh 4x4

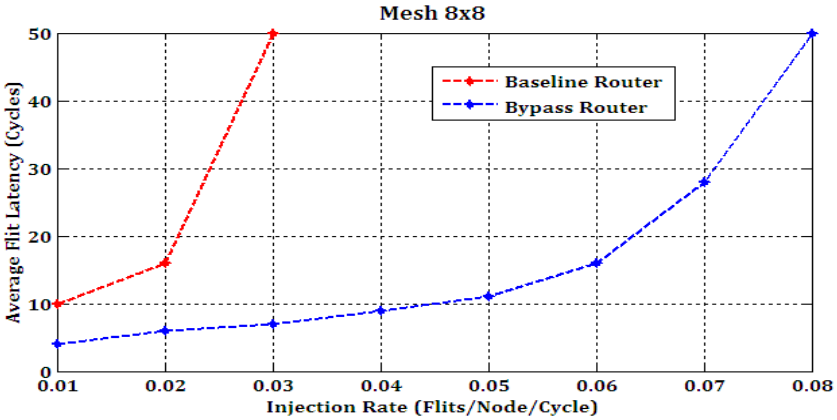
IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

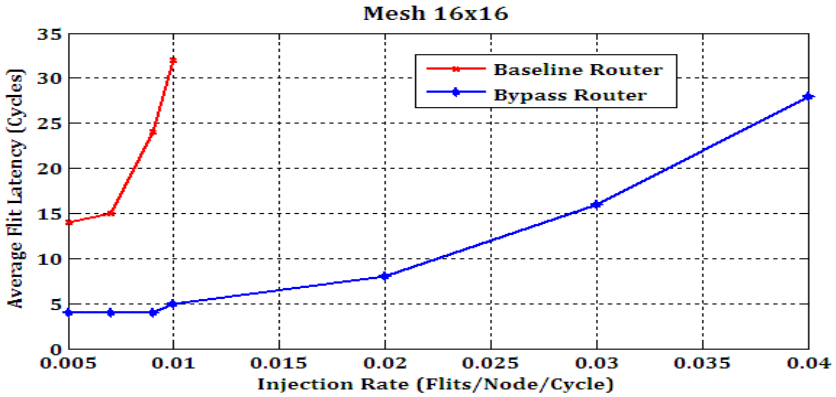
International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239



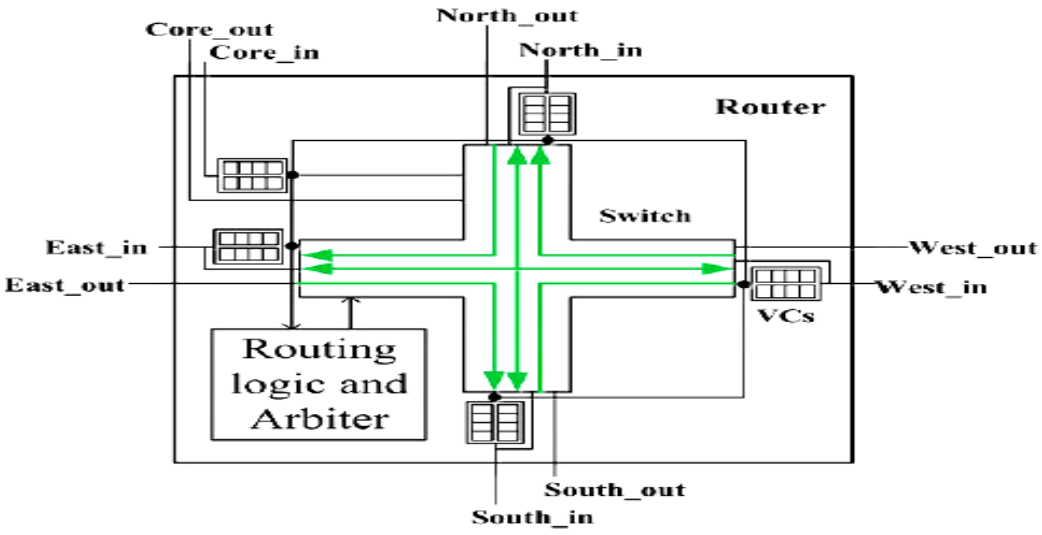
(b) Mesh 8x8



(a) Mesh 16x16

Figure.7. Bypass router with Butterfly traffic and proposed routing algorithm

Figure .8a shows a two-dimensional bypass router that is simulated in an 8 x 8 mesh using a semi-adaptive routing technique, and Figure 8b shows the latency decrease that results as a consequence of this simulation. Flits may then spin the router around and bypass both routers along the dimension they are working on (s). We compare the simulation results of a two-dimensional bypassing router to those of a baseline router that was also used in an 8-mesh environment and simulated using an adaptive routing method that included butterfly traffic. In the case of a base router, flits are trapped in each router, increasing latency; however, in the case of a bypass router, flits may be routed via the bypass path, reducing hop count (H) and latency.



(a) Turning to right Bypass

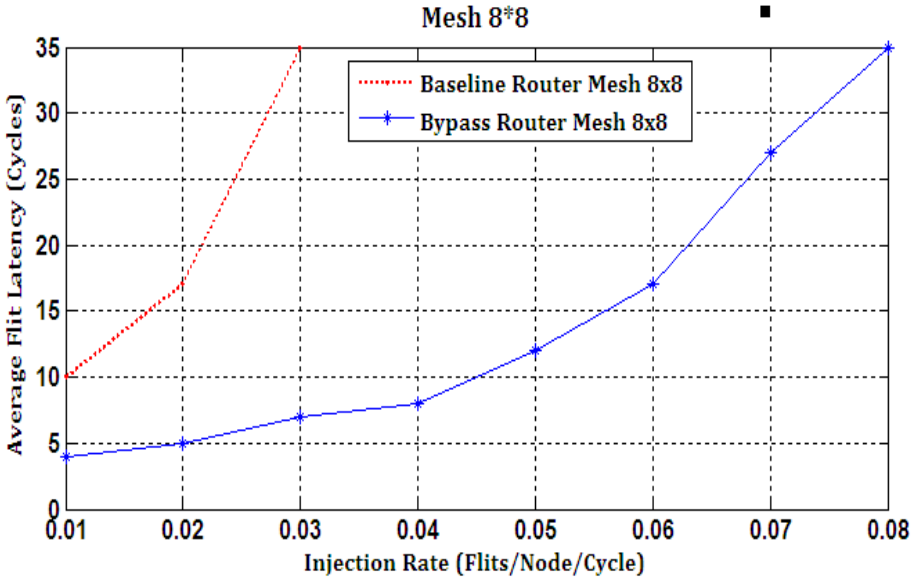
IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239



(b) 8x8 Mesh

Figure.8 Butterfly router uses a two-dimensional traffic bypass to provide dynamic routing, using a semi-adaptive algorithm.

Concerning an 8 x 8 mesh topology, the energy consumption of Smart is compared to that of a bypass router as well as that of the baseline router.

6. CONCLUSION

The paper proposes a novel flow control method for optimizing the power consumption, latency, and effectiveness of NoCs. Once a flit is received, its priority is determined and a suitable virtual channel (VC) is readied for assignment. The VCs are prepared using a customized RED algorithm. Following the arrival of a flit, it is possible to allocate some additional flits with much the same importance to that VC. Thus, increasing the VC following the entry of each flit is a prudent and preventative measure. To accomplish this modification, we combined the modified RED with a learning automata method. The program precisely changes the RED algorithm's thresholds based on network feedback. Because the VC size is dynamically adjusted, the buffer area is efficiently used, lowering the amount of packet loss. As a result, it lowers the number of packet retransmissions or long waits caused by insufficient space, as a consequence of which latency and power consumption have been significantly reduced. Latency is reduced by using the bypass router described in this article without the need for any physical control connections to be added to the network. In addition, we offer a new routing technique that takes advantage of the decreased latency in two-dimensional routes by attempting to flip the proposed router while doing so. According to simulation findings, configuring single cycle routes results in a substantial decrease in latency while also reducing power and area usage when compared to previous works' models. The findings indicated that precise buffer management may result in a reduction in power usage. But on the other hand, owing to a scarcity of buffer space There are certain instances when data resend is likely to happen, increasing the network's power usage. This additional power usage was included in all of the article's simulation tests. The proposed approach may be regarded as appropriate for systems that are prone to mistakes caused by a lack of adequate buffer space.

REFERENCES

[1] Krishna T et al (2013) Breaking the on-chip latency barrier using SMART. In: IEEE 19th international symposium on high-performance computer architecture (HPCA2013)

[2] Marculescu R et al (2009) Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives. Comput-Aided Des Integrated Circuit System IEEE Trans 28(1):3–21

[3] Grot B et al (2009) Express cube topologies for on-chip interconnects. High-Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th international symposium on IEEE

[4] Kim J, Dally WJ, Abts D (2007) Flattened butterfly: a cost-efficient topology for high-radix networks, ACM SIGARCH Computer Architecture News 35(2):126–137

[5] Kao Y-H et al (2011) CNC: high-radix closure network-on-chip. Computer-Aided Des Integrated Circuit System IEEE Trans 30(12):1897–1910

[6] Howard J et al (2010) A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS Solid-state circuits conference digest of technical papers (ISSCC), 2010 IEEE international, IEEE.

[7] Li M, Zeng Q-A, Jone W-B (2006) DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network-on-chip. In: Proceedings of the 43rd annual design automation conference, ACM.

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 12, December 2021

DOI: 10.17148/IJARCCE.2021.101239

[8] Jain TNK et al (2010) Asynchronous bypass channels: improving performance for multi-synchronous NoCs. In: 2010 Fourth ACM/IEEE international symposium on Networks-on-Chip (NOCS). IEEE

[9] Rodrigo S et al (2009) Efficient implementation of distributed routing algorithms for NoCs. IET Computer Digital Technology 3(5):460–475

[10] Matsutani H et al (2009) Prediction router: yet another low latency on-chip router architecture. In: IEEE 15th international symposium on high-performance computer architecture. HPCA 2009. IEEE, pp 367–378

[11] Kumar A, Peh L S, Jha NK (2008) Token flow control. In: Proceedings of the 41st annual IEEE/ACM international symposium on Microarchitecture. IEEE Computer Society.

[12] Jerger NE, Peh L-S (2009) On-chip networks. Morgan and c Laypool, Cambridge.

[13] Badri S, Holsmark R, Kumar S (2012) Junction-based routing: a scalable technique to support source routing in large NoC platforms. In: Proceedings of the 5th international workshop on Network on chip architectures, ACM

[14] Fazzino F, PalesiM, PattiD(2008) Noxim: network-on-chip simulator. http://sourceforge.net/projects/noxim

[15] Kim HJ, Park D, Theocharides T, Vijay Krishna N, Das CR. A low latency router supporting adaptivity for on-chip interconnects. In: Proceedings of the 42nd annual design automation conference (DAC); 2005. p. 559–64.

[16] Nayebi A, Meraji S, Shamaei A, Sarbazi-Azad H. Emulator: a listener-based integrated simulation platform for interconnection networks. In: Proceedings of Asian modeling symposium (AMS'07); 2007 Available: http://www.xmulator.com.

[17] Kahng AB, Li B, Peh L, Samadi K . ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration. In: Proceedings of Europe conference & exhibition on design, automation & test (DATE '09); 2009. p. 423–8.

[18] Kumar A , Peh LS , Kundu P , Jha NK . Express virtual channels: toward the ideal interconnection fabric. In: Proceedings of the 34th annual international symposium on computer architecture (ISCA'07), 35; 2007. p. 150–61.

[19] E.G. Satish, Sanju V, Srinivasa N, Navya Y.U, "A Research Review on Wireless Network on chip (WNoC)" in the international journal of computer engineering and applications volume IX, special issue IJCEA 2015

[20] V. Veeraprathap, M. Nagaraja and M. Z. Kurian, "Network on chip design and implementation on FPGA with advanced hardware and networking functionalities," 2019 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2019, pp. 1-6, DOI: 10.1109/ICCCNT.2019.6726748.

[21] U. R, S. H. and S. V., "Network-On-Chip (NoC) - Routing Techniques: A Study and Analysis," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-6, DOI: 10.1109/GCAT47503.2019.8978403.

[22] E.G Satish, Ramachandra A-C "Design of an Efficient Traffic Balance and Fault-Tolerant Routing Algorithm for Network on Chip". Design Engineering, ISSN: 0011-9342,1485-1500.http://www.thedesignengineering.com/index.php/DE/article/view/2148.2021

[23] E. Salminen, A. Kulmala and T. D. Hamalainen, "On network-on-chip comparison," 10th Euromicro Conference on Digital System Design Architectures, Methods, and Tools (DSD 2018), 2018, pp. 503-510, DOI: 10.1109/DSD.2007.4341515.

[24] S. Kumar et al., "A network on chip architecture and design methodology," Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2020, 2020, pp. 117-124, DOI: 10.1109/ISVLSI.2020.1016885.