

An Energy-Efficient and Reliable Asynchronous Messaging Framework For Multi-Core SoCs

G. Munirathnam^{*1}, Y. Murali Mohan Babu²

¹Assistant Professor in ECE Department, JNTUACEK, Kalikiri, Research Scholar, JNTUA, Ananthapuramu, A.P.

²Professor in ECE Department, NBKRIST, Vidyanagar, A.P.

Abstract

In the modern era, multi-core technology is one of the effective communication between the various IP-enabled cores and it is a critical challenge. As more and more IP cores are integrated into multi-core system-on-chip (M-SoC), traditional bus architectures are inadequate to meet the performance demands of modern systems. Also, interconnection architecture for on-chip multi-core communication has an increasing presence in research due to its scalability and efficient communication. M-SoC is the integration of multiple processing cores to implement the communication performance effectively within the SoC. Consequently, conventional messaging protocols fail to optimize the clock distribution challenges and imperfect fault tolerance. To achieve better performance, adaptive routing algorithms need to monitor network status and make routing decisions informed by congestion data, which unavoidably increases design complexity and overhead. Therefore, this paper developed an Adaptive Fault Tolerant with Hybrid Congestion Aware routing protocol (AFT-HCARP) is designed to improve the performance while ensuring the asynchronous communication protocol. Here, a deterministic-adaptive hybrid routing strategy was utilized to reroute the transmitted messages around the overloaded modes. In addition, the proposed AFT-HCARP protocol is integrated with dynamic routing adaptation as well as a destination-based adaptive routing algorithm to validate the finest message transferring under several network conditions, which are varied under the fault scenarios. Moreover, the experimental outcomes were verified in terms of throughput performance under various fault conditions, latency to load, and finally comparative analysis is performed across the baseline protocols. Furthermore, the results show that developed AFT-HCARP reduces average packet delay as 3.8cycles and enhances the throughput as 0/0098 flit/node/cycle. From the comparison, fault recovery time is 4 cycles, and packet delivery ratio is 97.60% than the conventional routing schemes.

Keywords: Multi-Core Technology, Processing Cores, Imperfect Fault Tolerance, Adaptive Routing Algorithms, Hybrid Routing Strategy, Dynamic Routing Adaption

1. Introduction

The standard communication medium for multi/many-core systems on-chip was done by Network-on-chip (NoC) because of its various properties including parallelism and scalability [1]. Topology, microarchitecture, routing algorithms and switching strategies are the factors that influence the performance of NoC [2]. Since now we rely more on data for decision-making and computing, there is a growing need for small, powerful and energy-efficient chips that can work at the edge of network is closer to where the data is created. These edge devices help to reduce the delays and ease the load on centralized data centres, which often struggle with limited bandwidth and high latency [3]. Embedded systems, powered by compact and efficient System-on-Chip (SoC) designs, can be made more sustainable by reducing size and energy use through the Softwares like real-time operating systems, smaller processors for simple tasks and open-source components which allow for greater transparency and optimization [4]. This work presents fault-tolerant asynchronous messaging protocol for multi-core Systems-on- Chip (SoCs), addressing growing reliability challenges caused by hardware complexity and radiation effects by using N- Modular Redundancy (NMR) and division to ensure dependable communication [5]. This work studies how different controller designs inside the communication part of MSoCs can make them more reliable and work better for critical systems [6].

Inspired by systems like SpiNNaker that uses multi-core chips with built-in fault protection and smart communication, a flexible and reliable messaging method is introduced to help multi-core chips work better, especially for real-time and large-scale tasks [7].

As MSoCs are used in more complex and sensitive systems, it's important to make sure they can communicate safely and keep working even when something goes wrong, that helps SoCs stay reliable and adapt to different needs [8]. In systems where devices work together wirelessly, it is important to have a way for them to agree on decisions even if some parts fail or the connection is unstable [9]. Reliable operation of induction motor drives in critical applications requires adaptive fault-tolerant control systems that detect sensor faults and enable smooth transitions between control methods, supported by real-time IoT-based condition monitoring [10]. Fault-tolerant control framework for semi-Markov jump systems incorporates random communication delays and non-smooth sensor faults to improve control performance and reduce network usage [11]. Multi-agent systems widely applied in fields like power grids and robotics require fault-tolerant and event-triggered consensus control [12]. The distributed adaptive event-triggered fault-tolerant cooperative control is developed for heterogeneous multi-agent systems for ensuring stability and bounded tracking errors [13]. Modern electronic systems require high reliability and predictability, especially in fields like medicine and aerospace, motivating ongoing research into time-predictable architectures and fault-tolerant designs that ensure continuous, error-free operation even under harsh environmental conditions [14].

In previous several studies were done to handle the fault tolerance mechanism in an effective way. The conventional protocols are may lack detailed experimental results or practical implementations [16]. Combining deterministic and adaptive routing may increase design complexity compared to purely deterministic schemes [17]. Static optimization technique [18]. Therefore, this article aims to deal the NoC reliability, congestion management and fault tolerance model. Here, a hybrid routing protocol is used to handle the functions effectively. Moreover, the key contribution of the developed AFT-HCARP protocol is summarized in below,

- Initially designed a novel AFT-HCARP protocol to overcome the fault tolerance challenges and communication overheads from the M-SoC.
- The, asynchronous communication modules were integrated with proposed protocol to enhance the system reliability and control the faults during the transmission time.
- After that, to avoid overlapped nodes and dynamic rerouting an adaptive and deterministic routing models are integrated.
- Here, congestion level, faulty nodes and network levels are analysed to apply the destination based dynamic routing logic with many fault conditions under the NoC environment.
- Finally, the model is inverted to implemented depends on the network parameters like higher PDR and throughput, less average packet delay and fault recovery time.

The research article was organized in the order as follows. Section 2 is a summary of the existing work, and issue formulation is found in Part 3. The hybrid routing method is presented in Section 4. The results of the simulation on the test framework are displayed in Section 5. finally, the paper is concluded with Section 6 conclusion.

2. Related works

The research [19] concluded by outlining an adaptive two-way routing technique for hotspots in NoC multiprocessors. The algorithm improves upon conventional routing

techniques by lowering packet latency by detecting hotspots and implementing route limitations. The experimental analysis proves that the proposed technique is effective in reducing congestion even when dealing with hotspot cores.

Reference [20] developed a routing algorithm for NoC architectures. The algorithm aims to efficiently distribute on-chip traffic across the network by using global traffic information rather than relying solely on local congestion information. It accomplishes this by employing a lightweight and efficient control network that monitors on-chip traffic and collects the necessary global information to determine appropriate paths between communicating nodes. The proposed method is demonstrated to outperform conventional deterministic and adaptive routing mechanisms in terms of congestion avoidance and load balancing based on experiments conducted on a 7x7 NoC using several benchmarks.

The paper [21] introduced an innovative methodology to tackle the problem of adaptiveness in NoC routing. The proposed approach enhances network performance by introducing additional routing options and distributing traffic load in non-congested regions through the inclusion of cycles in the channel dependency graph, subject to specific constraints.

This paper [22] emphasized the importance of addressing system level reliability concerns within the emerging NoC paradigm for MP-SoCs. The proposal suggested the integration of adaptive into data communication systems in order to enhance their ability to withstand manufacturing faults. This paper emphasized the importance of a comprehensive study on trade-offs in performance and cost related to adaptive routing schemes used with NoC fabrics, considering application-specific requirements. Consequently, the research gap of the existing works are summarized as table.1.

Table.1 Research gap

Sl.no	Developed techniques	Pros	Cons	Outcomes
1	adaptive two-way routing technique [19]	effective in reducing congestion even when dealing with hotspot cores	Effective congestion reduction rates over the hotspot cores.	Higher PDER and FDR under the network conditions
2	routing algorithm for NoC architectures [20]	outperform conventional deterministic and adaptive routing mechanisms in terms of congestion avoidance and load balancing based on experiments	Recovery challenges are not mitigated	Less power consumption and bandwidth
3	innovative methodology [21]	It can manage both soft and hard faults from the adaptive routing models.	In some cases complex to design the protocol as the circuit level verification status	Low latency and packet drop
4	NoC paradigm for MP-SoCs [22]	Dynamic routing detection	It can introduce power	Higher performance

		performance can make the system very effective and avoid the faulty nodes	overhead across the SoC constraints	under PDR and FDR
--	--	---	-------------------------------------	-------------------

3. Motivation

The existing protocols has several drawbacks based on static routing in terms of complex adaptive schemes. This has included latency as well as design overheads to ensure the fault resilience in the asynchronous communication towards the non-integrated routing models [23]. Also, in chip interconnection networks uses NoC design model but it does not manage the faults with respect to the static path. This model has adapted in deterministic routing path to lacks the fault recovery performance and a integration of the NoC design [24]. Distributed congestion aware protocol is type of energy aware routing strategy to ensure the complexity performance during the M-SoC management. However, it does not support and applicable for fault prone concept and congestion management.

4. Proposed methodology

The M-SoC switching elements and configurations of the developed AFT-HCARP is illustrated in fig.1. Main of this techniques is to optimize the routing performance based on the fault conditions and various congestion control 5G networks. Consequently, the developed model contains three main blocks such as top, middle and bottom. These blocks are having various performance for responsibilities for network interface and functions of switching elements are done within the top block. Routing operations and their transportation functions are done using the middle blocks. Then, adaption as well as dynamic routing performance were done on bottom section. Moreover, the transport layer has integrated the deterministic and adaptive hybrid strategy to ensure the routing performance is well suitable for identifying the faulty paths. The asynchronous FIFO was used to reroute the transmission and temporarily holds the data packets while having any faulty nodes. Also, it has provides reliable and synchronization at destination.

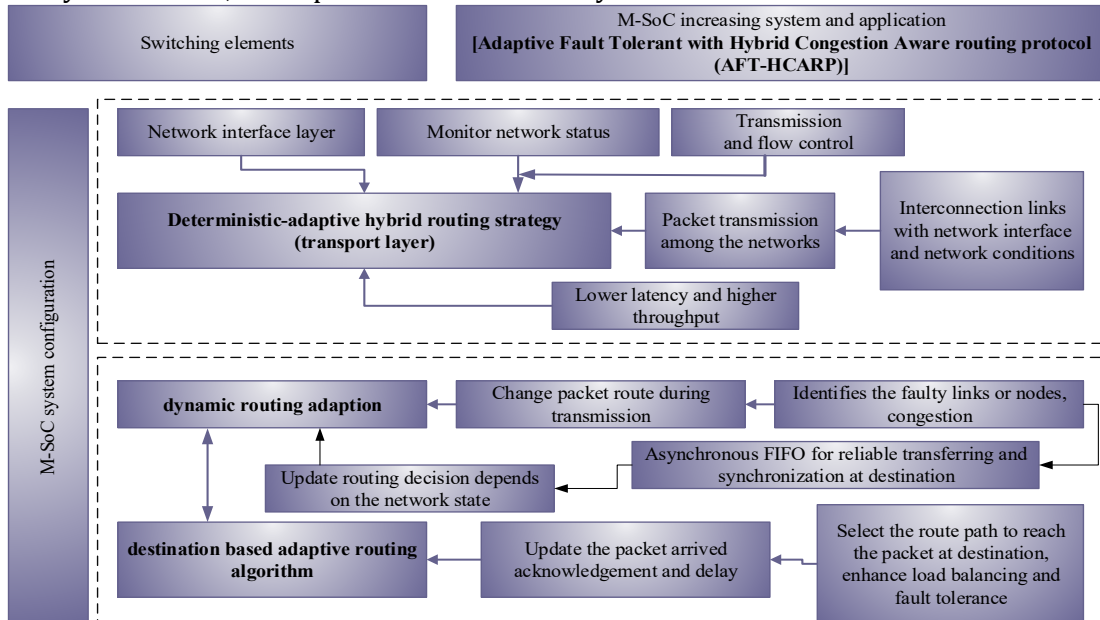


Fig.1 Proposed architecture

Then, ensures the all the packets are transmitted between the networks and selected routes. Also, current network status was validated under the interconnection link among the logical transmission nodes. Routing decisions are updated based on the node

availability, status of network under congestion levels. The main of this protocol is to enhance the transition process faster and without delay performance.

4.1 Deterministic-adaptive hybrid routing strategy

Deterministic routing strategy is used to ensure the way of data packets also, which has validated under mapping conditions like predetermined, predictable and consistent network. Moreover, the conventional routing strategy can change the network traffic based in the specific configurations. Therefore, this strategy is well suitable for enhancing security and NoC communications. Also, the deterministic router structure consists virtual channel controllers and crossbar switches which are placed x and y directions in terms of 2D mesh NoC topology, which is demonstrated in fig.2. This router function validates the data packet transferred along the pre-defined network paths.

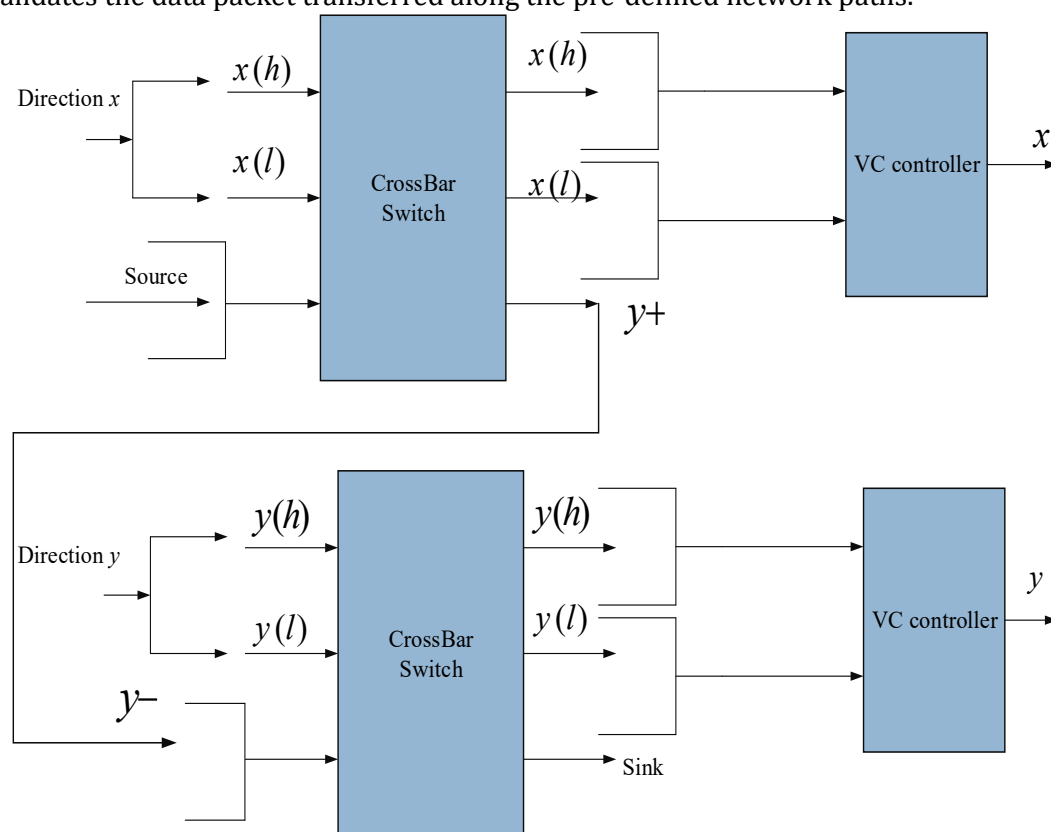


Fig.2 Deterministic router

From the diagram represents each data packets are delivered from the neighbouring routers in the x and y directions. Then, the data packets are injected into the network with the help of local cores. From the diagram is denoted as higher priority of virtual channels and is represented as lower priority of virtual channels for both the directions x and y . Then, multiple input and output are connected in the crossbar switch with higher bandwidth. Also, this crossbar switch, has validates the inaccurate or non-blocking connection among the source to destination with x and y directions. Here, this point the output function is mainly based on the routing logics and crossbar routes for higher and lower priority of virtual channels.

Initially, the data packets are transferred along with the x direction till it arrives the accurate path. Then, this data packets are transferred along with the y direction to arrive the destination. Here, $x(s')$ and $y(s')$ is the source node representations and

$x(d')$ and $y(d')$ is the destination node representations. In addition, x direction is assumed as horizontal direction and y direction is assumed as vertical direction. Then apply the routing decision at both horizontal and vertical direction using eqn. (1) and (2),

$$x = \begin{cases} x^+ & \text{if } x(d') > x(s') \\ x^- & \text{if } x(d') < x(s') \end{cases} \quad (1)$$

$$y = \begin{cases} y^+ & \text{if } y(d') > y(s') \\ y^- & \text{if } y(d') < y(s') \end{cases} \quad (2)$$

Where, the eqn. (1) works based on the condition as $x(s') \neq x(d')$, similarly the eqn. (1) works based on the condition as $x(s') = x(d')$. The deterministic router uses storage buffers associated with output channels, while the adaptive router uses storage buffers associated with input channels. When using output buffers, the routing decision must be made before buffering the message which is ideal for deterministic routing since only one choice is available for an incoming message. Then, apply the virtual channel using the allocation logic in eqn. (3),

$$V(c)_o = [R(s)_i, P_d] A_f \quad (3)$$

Where, A_f is denoted as function of arbitration, input request signal is represented as $R(s)_i$ and is expressed as P_d . When using input buffers, the routing decision must be made after buffering the message. It is suitable for adaptive routing since a message can usually be routed on several possible output channels. The adaptive router implements a round-robin input message selection policy which checks for messages first among all adaptive buffers and then among all deterministic buffers. Crossbar selection function $C(s)_o$ is mentioned in eqn. (4),

$$C(s)_o = M_c \times V_i(a) \quad (4)$$

Where, M_c is represented as mapping matrix in terms of crossbar selection and $V_i(a)$ is symbolized as vector functions of active inputs. Output channel selection is performed by giving priority to those channels in the dimension with the greatest number of hops remaining for the selected message. Each dimension with decreasing number of remaining hops is tried until a free channel is found or all channels have been tried. By using this output channel selection policy, the greatest amount of adaptively for a message is retained which reduces blocking. Finally, validation of buffer availability is mentioned in eqn. (5),

$$G(v) = \begin{cases} 1 & \text{if } BV(c) > T \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where, $BV(c)$ is denoted as availability buffer in virtual channels from both x and y directions and T is characterized as threshold function.

Fig.3 represents the adaptive router with 2D mesh NoC architecture which is extension of deterministic routing strategy. Moreover, this router can effectively utilize the congestion control and traffic information to the corresponding direction such as x and y . Here, $x(a)$ and $y(a)$ is denoted as higher and lower adaptive priority of virtual channel from both directions x and y .

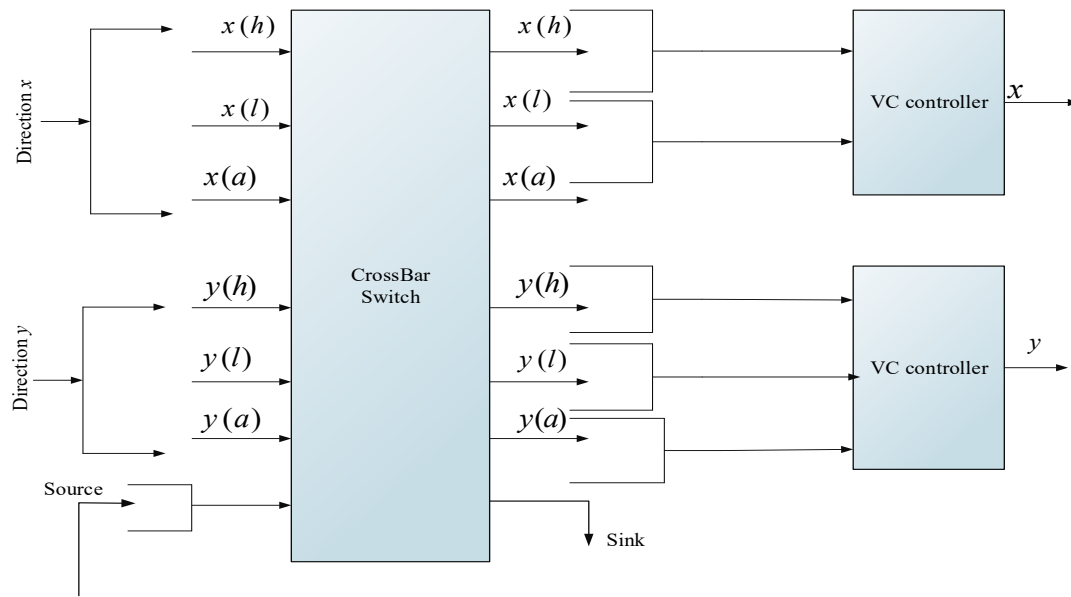


Fig.3 Adaptive router

Consequently, flow control digits or flits are injected into the network with those local cores. These flits are again transferred to the crossbar switches, which has mapped the input to outputs. In that adaptive routing, non-blocking communications are allowed to adaptive switching performance for routes the input data packets from any other input port to output port with their buffer availability. From the adaptive routing congestion metrics are added for x and y such as $x(c)$ and $y(c)$. Here, apply the outing decision towards the lesser congestion using eqn. (6),

$$S_p = \begin{cases} x(d') & \text{if } x(c) < y(c) \\ y(d') & \text{otherwise} \end{cases} \quad (6)$$

From this eqn. (6) congestion metric is similar, this is drop back to deterministic routing strategy. Then, select the output port using eqn. (7),

$$P_o = \underset{p \in (x', y')}{\arg \min} \rightarrow p(c) \quad (7)$$

Where, $p(c)$ is denoted as buffer occupancy which is estimated based on the flit queue length. After estimation, of buffer occupancy virtual channels are allocated to each routing decision which is an output of virtual channel $V(c)_o$ using eqn. (8),

$$V(c)_o = A_f [R_d, B_s, V(c)_a] \quad (8)$$

Where, R_d is denoted as routing decision, B_s is represented as buffer status and $V(c)_a$ is expressed as availability of virtual channels. Finally, the data packets are transferred for the local nodes and this can be routed under the sink node. Consequently, the adaptive routing algorithm identify the packet coordination and validates it with the current router, then forward the processing acknowledgement it the local core. Here, fault tolerance is reduced and enhanced the load balancing performance.

4.2 Asynchronous FIFO Mechanism

Asynchronous FIFO has consisted four main stages such as input stage, adaptive router, routing decision stage and finally output stages. Consequently, input stage involves various IP cores which are transfers and receives the data packets through the FIFO. Here, the FIFO was enabled at each input path of the router to decouple the clock domains.

Then, Deterministic-adaptive hybrid routing strategy was applied as the routing computation unit to detects the overlapping paths and verify the faulty nodes across the packet transferring paths. In addition, routing decisions such as normal conditions and fault detection conditions are stored in the fault recovery buffer. Fig.4 shows asynchronous FIFO with adaptive routing model.

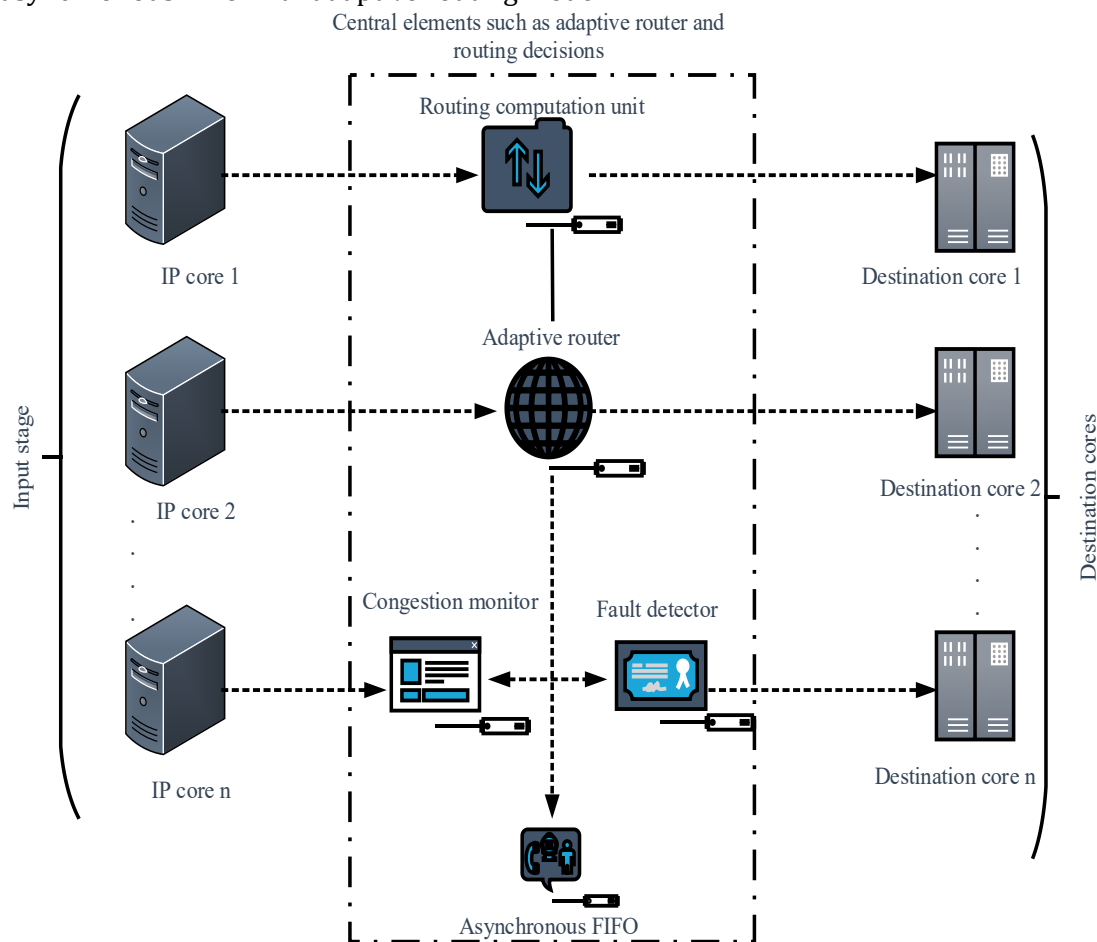


Fig.4 Asynchronous FIFO architecture

Moreover, M-SoC can combines various processing core to improves the performance such as communication delay, fault tolerance and throughput within the asynchronous FIFO input buffers. Here, this module has effectively managed the data transferring among the heterogeneous cores for enabling the reliable communication. After that, the transferred packets are reached to the destination cores and finally, this has provided smooth data managing performance across the various clock signals.

4.3 Dynamic routing adaption as well as destination based adaptive routing algorithm

The suggested routing approach incorporates destination-based adaptive algorithms and dynamic routing adaptation into a deterministic-adaptive hybrid model. Packets first travel along deterministic routes that are determined by the relative locations of the source and destination nodes. However, the system dynamically modifies the routing path if network congestion, errors, or buffer overflow are identified during transmission. By analysing congestion measures in both possible directions (x and y), it makes judgments in real time while constantly taking the destination node's relative position into account.

This guarantees intelligent packet rerouting that strikes a balance between delivery accuracy, fault tolerance, and efficiency. In addition to improving delivery metrics like latency, throughput, and delivery ratio, this dual-level adaptation strengthens the network's resilience.

• **Traffic patterns:**

Moreover, the performance AFT under each non-random traffic patterns such as bit reversal and bit shuffle. Consequently, hybrid router has performed better at lower traffic patterns, while the AFT slightly better at higher traffic pattern both FDR and PDR. This has been done on deterministic router with higher priority *x* and *y* direction. In case lower selection is adapted in routed message through the deterministic channels.

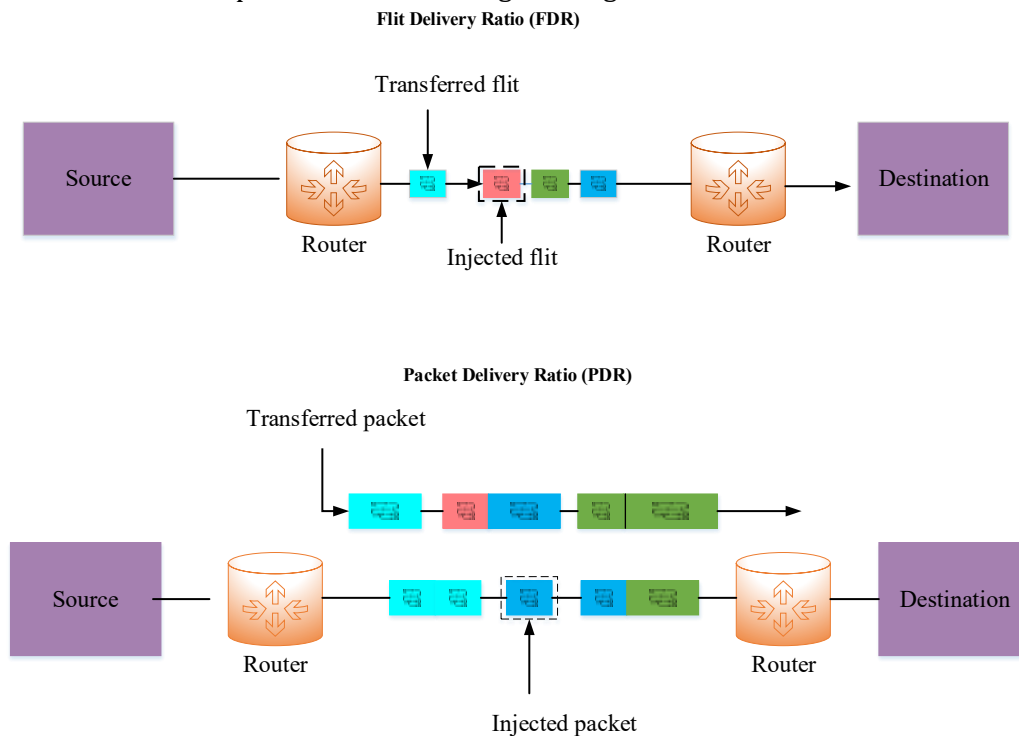


Fig.5 Transmission of packets and flits

From the fig.5 illustrates the message transmission on M-SoC with overloaded nodes representation. Here, the network router is starts with deterministic routing with low priority directions. So, the congestion nodes and faults nodes are identified through the alternate path to switches the adaptive routing module. Also, during the congestion the missed packets are again rerouted dynamically through the paths. This performance has enhanced the FDR and PDR functions under the source to destination paths. Moreover, adaptive router delay performance is derived under three main parameters such as $t'(r)$, $t'(s)$ and $t'(c)$. Initially estimate the total time taken to route the packets, which is denoted as $t'(r)$, and eqn. (9)

$$t'(r) = t'(ad) + t'(ea) + t'(hl) \quad (9)$$

Where, $t'(ad)$ is represented as delay time with respect to decoding the packets also, this is to analysing the packet header and designing the new packets header for all selected routers, $t'(ea)$ is denoted as delay time with respect to arbitration of routing function and $t'(hl)$ is expressed as delay time with respect to header selection from the

delivered packets. After that, find the time required to deliver flit to the particular output channel which is denoted as $t'(s)$ and eqn. (10),

$$t'(s) = t'(fc) + t'(cb) + t'(l) \quad (10)$$

where, $t'(fc)$ is denoted as delay time with respect to flow control for x and y direction, $t'(cb)$ is represented as delay time with respect to crossbar and $t'(l)$ is explored as delay time with respect to latch function. Then, derive the equation for $t'(c)$ in eqn. (11),

$$t'(c) = 4.9 + t'(vcc) \quad (11)$$

Where, $t'(c)$ is denoted as time taken to deliver a flit or packets across a PC and $t'(vcc)$ is denoted as delay time with respect to virtual channel controller. This includes multiplexing virtual channels into the physical channels. Moreover, the router performance is simulated based on degree of freedom in an each delivered packets and flits.

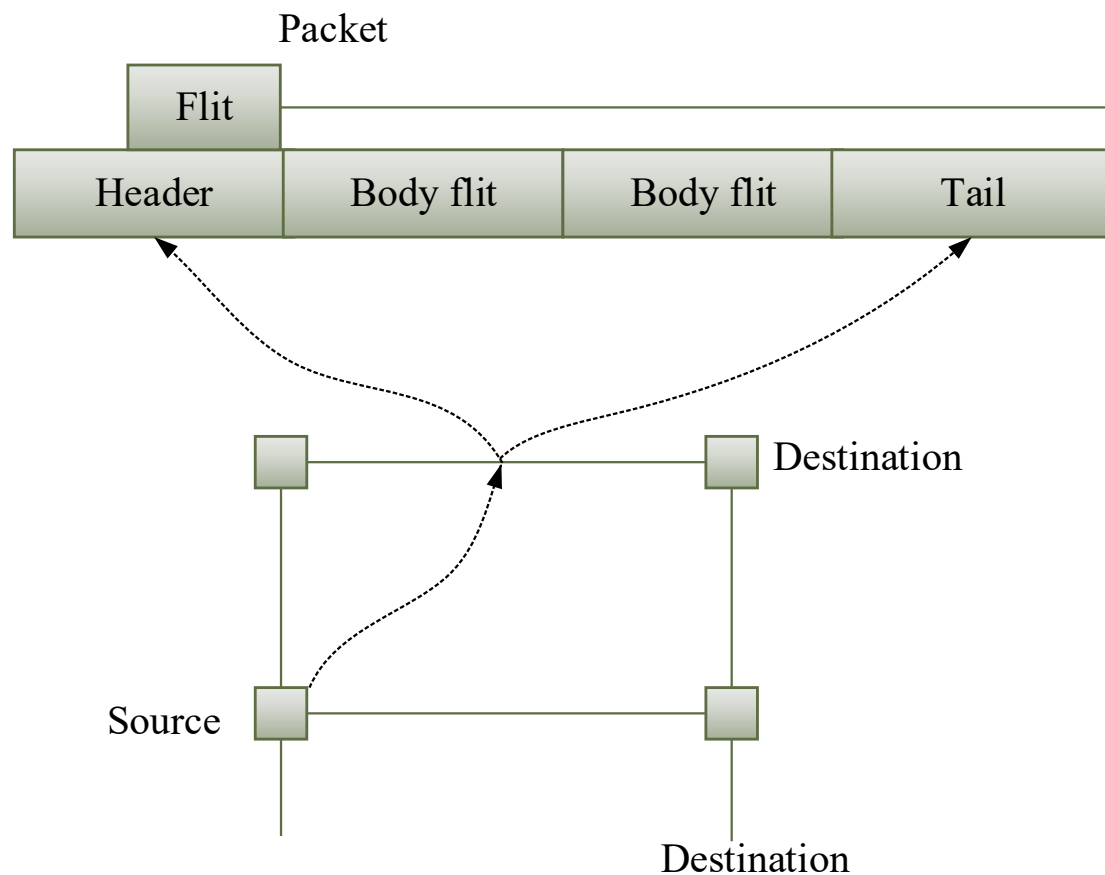


Fig.6 Packet Format

In an M-SoC system, the fig.6 shows that the flits or packets are properly injected and stimulated among the two nodes such as source and destination. By the rerouting transportations flits/packets are around loaded and routers, such as Deterministic-Adaptive Hybrid Routing Strategy enabled to improve both FDR and PDR performance over the networks. By switching to the adaptive pathways, network status can make data efficiently transmitted to the paths which are not congested and to make delivery reliability without packet loss. Consequently, packet drops made up of numerous flits, which consists header, body, and tail. Also, 2D mesh topology with a source node and destination nodes is then designed to overcome the network conditions like buffer

residence. 2D mesh topology are used by the router to dynamically select a network path. If a delivered network path is broken, it can reroute packets.

- **Congestion aware:**

The congestion of the communication infrastructure of an M-SoC depends on several factors, which, in the case of NoC, is closely related to the amount and type of traffic, latency, and network throughput. In addition, the characteristics and properties of routing and arbitration schemes play an important role in network congestion. Therefore, monitoring various metrics can improve network performance, such as leveraging information from buffers, which allows dealing with dynamic traffic loads through cognitive processes and control techniques [10]. Another improvement is identifying data flows that congest the network in certain areas or situations and subsequently avoiding them, resulting in considerable energy savings

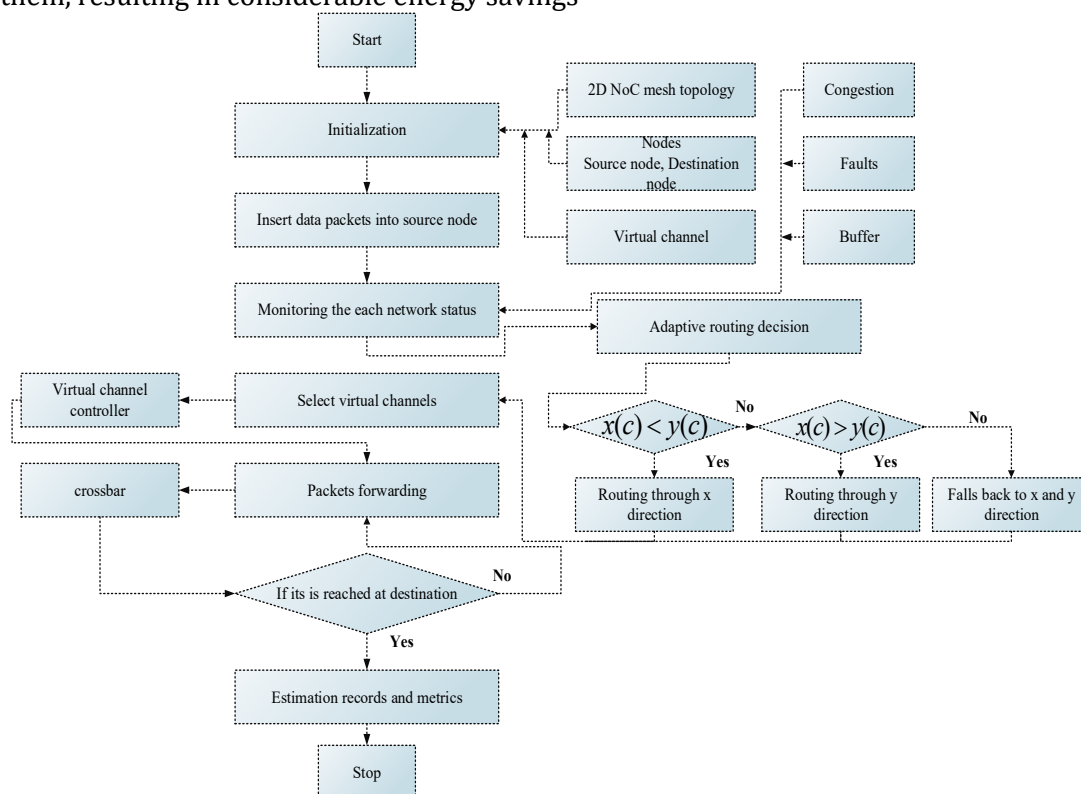


Fig.7 Flowchart for proposed methodology

Initially, the process is starts with routing execution with initialization of 2D mesh topology and define the elements such as virtual channels, source and destination nodes. Also, network parameters like buffer and fault status, congestion level. Then, the packets are converted into the flits, which are inserted into the source router. After that, network status is monitored if there are any network independency like buffer occupancy levels and congestion at routers. After that validation, adaptive routing decision can provide the decision according to the x and y direction. Here, itself faulty nodes and overloaded networks are rejected to enable the smooth process. Decision is implements and virtual channel is to forward the packets with the presents of virtual channel controller. Finally, an optimized fault tolerant data transmission process is done from source to destination using that developed AFT-HCARP model which is mentioned in fig.7.

5. Result and discussion

Simulation of the developed AFT-HCARP protocol execution were processed using the discrete-time simulator with 8-ray 3-cube network. Moreover, this simulation utilizes the stabilization threshold is 0.005 difference among 1000 clock cycles traffic with steady state performance. Consequently, the packets sizes are differentiate among the 8 to 32 flits under saturation traffic pattern 0.1. Moreover, the developed routing protocol can used three virtual channels per dimension and this packets length has equal buffer size during the simulation interval.

5.1 Case study

Let as assume $t'(ad) = 2.7$ $t'(ea) = 0.7$ $t'(hl) = 0.7$ $t'(fc) = 0.8$ $t'(cb) = 0.7$ $t'(l) = 0.7$ $t'(vcc)$ here, these values are substituted into eqn. (9), (10) and (11),

$$t'(r) = 2.8 + 0.7 + 0.6 * \log_2 f' + 1.5 + 0.6 * \log_2 f' \quad (9)$$

$$t'(s) = 0.7 + 0.6 * \log_2 b' + 0.5 + 0.6 * \log_2 p' + 0.9 \quad (10)$$

$$t'(c) = 4.9 + 1.25 + 0.6 * \log_2 c' \quad (11)$$

From these eqn. (9), (10) and (11) we substitute $f' = 1$, $c' = 2$, $b' = 1$ and $p' = 3$ finally got the determined values is $t'(r) = 5.0$, $t'(s) \approx 3.05$ and $t'(c) = 6.8$.

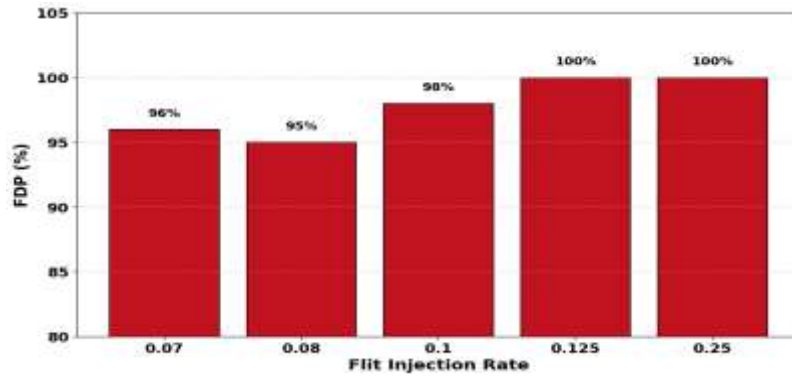


Fig.8 FDP under bit reversal traffic pattern

Fig.8 demonstrates FDR under bit reversal traffic pattern against the flit injection rate which is varying under the flit loss. From the demonstration shows FDR performance is increased with respect to the flit injection rate. Moreover, the lower flit loss can FDR attains 100% indicate the faster delivery performance from 0.07 to 0.10. Consequently, 100% at the 0.125 flit injection rate which is finest performance under bit reversal traffic pattern

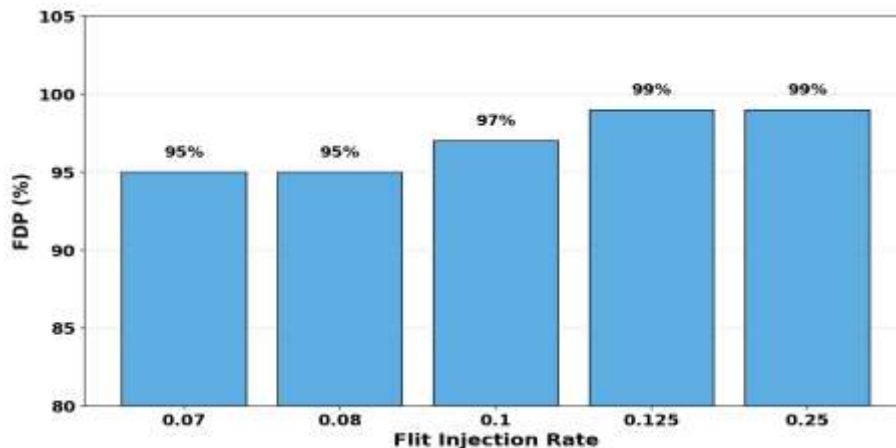


Fig.9 FDR under bit shuffle traffic pattern

Fig.9 illustrates FDR under the bit shuffle traffic pattern, which indicates the developed model has better performance by varying the flit injection rate with enhancement. Moreover, the flit injection rate is starting from 0.007 to 0.25 which measures the reliability of flow control units according to the network. Here, the lesser injection rate indicates flit loss and have some delay. To increase the injection rate FDP is exponentially increase up to 90%.

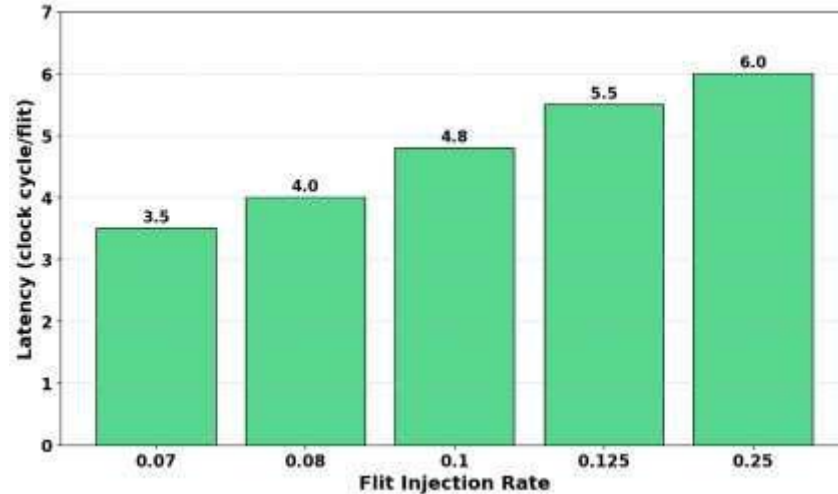


Fig.10 Latency per channel under bit reversal traffic pattern

Fig.10 demonstrates latency per channel under bit reversal traffic pattern for various injection rates. Here, the latency performance has measured in terms of clock cycles per flit which shows the network is the corresponding path to deliver the flits through that. At 0.07 injection rate suggests lower efficient for both delivery and congestion for 3.5 clock cycles per flit. Moreover, to increase the injection rate latency also increases rapidly from 4.0 to 4.8 clock cycles per flit. At last, 5.5 to 6.0 clock cycles per flit latency have achieved 0.25 to 0.07 injection rate respectively. From the observation, by varying the flit injection rate latency also increases due to the highest network utilization as well as congestion.

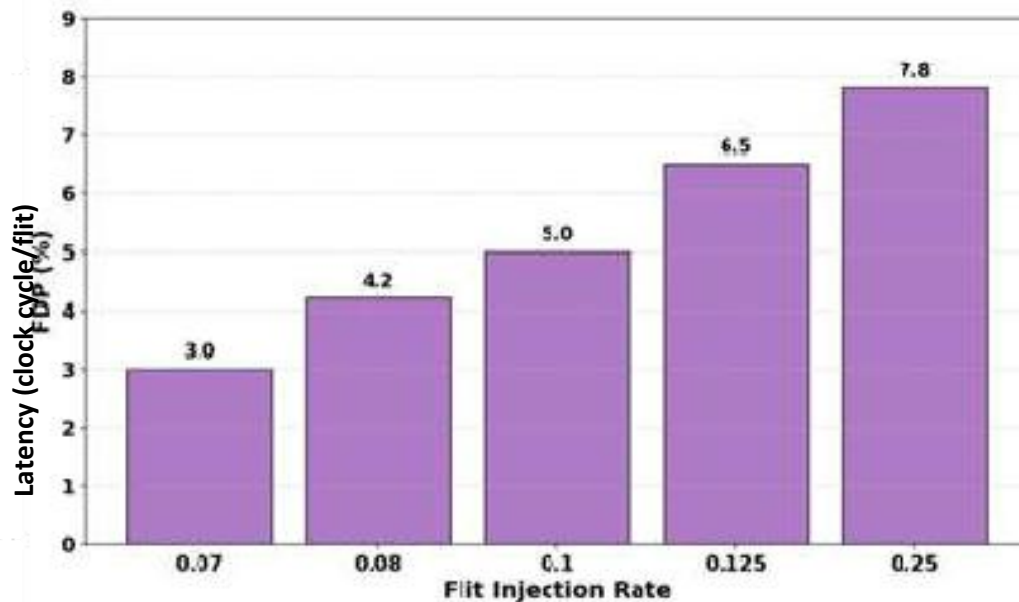


Fig.11 Latency per channel under bit shuffle traffic pattern

Fig.11 demonstrates latency per channel under bit shuffle traffic pattern for various injection rates. This bit shuffle traffic pattern demonstrates at increasing the bit injection ratio latency also increased which indicates the developed AFT-HCARP protocol has higher network loads.

5.2 Performance assessment and comparison

The implementation work of the developed AFT-HCARP protocol is done by the discrete-time simulator with 8-ray 3-cube network and the parameters like throughput, energy consumption, delay packet loss, and packet transmission are calculated. Moreover, the developed approach is validated using existing routing protocol like First Deterministic 3.5D Network Topology (FDNT) [25], Task Mapping Algorithm (TMA) [26] and Hybrid application Mapping Routing (HAMR) algorithm [27].

5.2.1 Packet Delivery ratio (PDR)

The developed AFT-HCARP protocol has attained lower value of packet drop which indicates this has optimal PDR over the networks. Moreover, PDR is estimated to find recognize the total amount of packet delivering in the entire network. In addition, the ratio of delivered packets (D_p) and received packets (R_p), which is mentioned in eqn. (12)

$$PDR = \frac{D_p}{R_p} \quad (12)$$

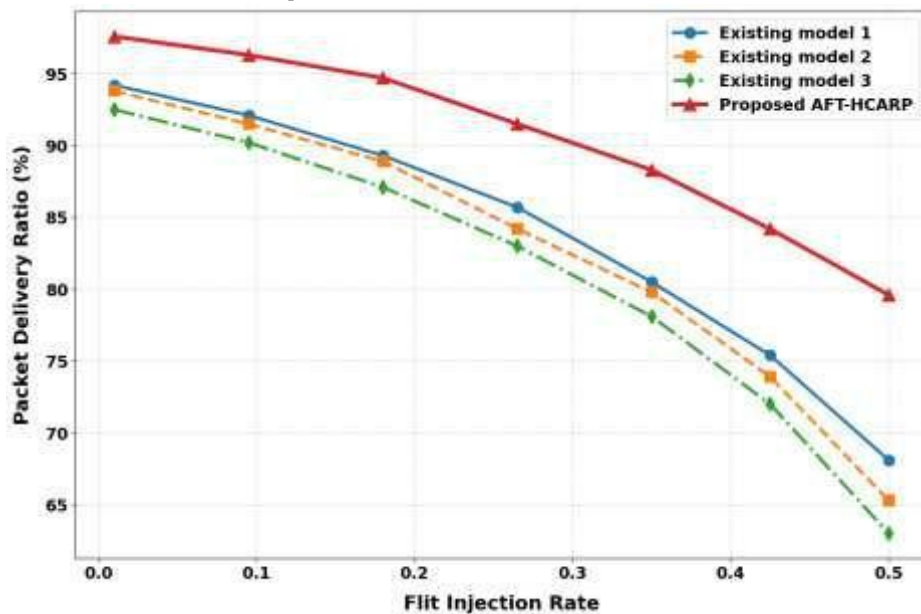


Fig. 12 comparison of Packet delivery ratio

Here, the PDR analysis of the developed AFT-HCARP protocol performance is compared with traditional routing algorithms such as FDNT, TMA and HAMR, which are demonstrated in fig.12. Initially, FDNT protocol has attained 94.20%, 92.10 %, 89.30%, 85.70%, 80.50%, 75.40% and 68.10% PDR rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. Similarly, TMA has reached 93.80%, 91.50%, 88.90%, 84.20%, 79.80%, 73.90% and 65.30% PDR rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. In addition, HAMR has achieved 92.50%, 90.20%, 87.10%, 83.00%, 78.10%, 72.00% and 63.00% PDR rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. But the developed AFT-HCARP protocol significantly

outperformed as 97.60%, 96.30%, 4.70%, 91.50%, 88.30%, 84.20%, 79.60% various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 while comparing the traditional models.

5.2.2 Fault recovery time

Fault recovery time is denoted as total time taken to process the network system such as router and M-SoC systems to identify normal communications, faults nodes and retransmission of data packets through the network path. According to the developed protocol fault recovery time calculated using in eqn. (13),

$$FRT = \frac{1}{n(f)} \sum_{j=1}^n (t_i(r) - t_i(f)) \quad (13)$$

Where, $n(f)$ is denoted as observed fault events from the path, $t_i(r)$ is represented as time taken the fault detection and $t_i(f)$ is expressed as time taken the resend packets over the path.

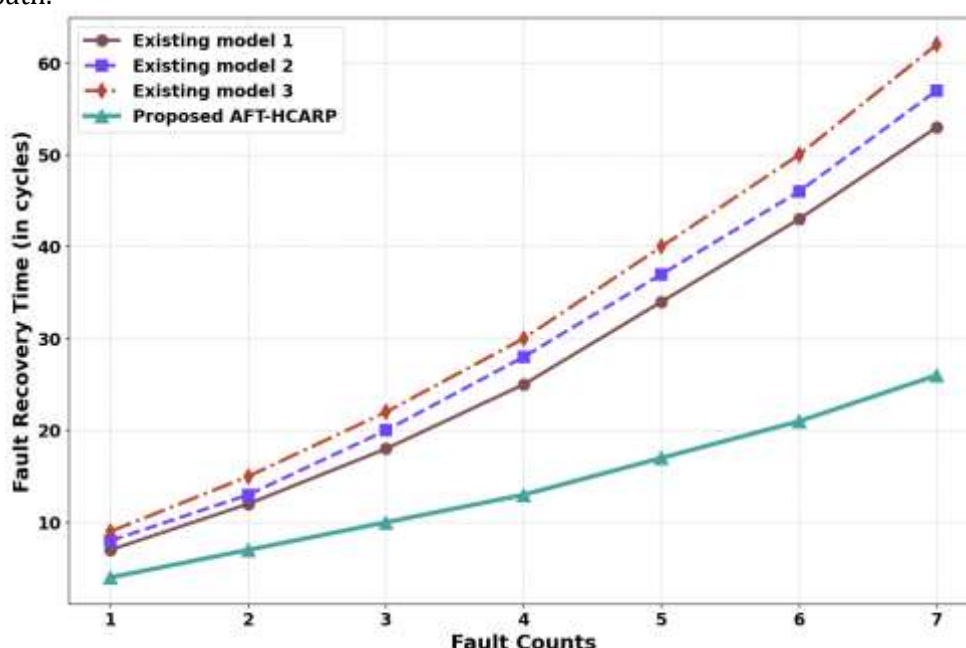


Fig. 13 comparison of fault recovery time

Here, the fault recovery time analysis of the developed AFT-HCARP protocol performance is compared with traditional routing algorithms such as FDNT, TMA and HAMR which are demonstrated in fig.13. Initially, FDNT protocol has attained 7, 12, 18, 25, 34, 43 and 53 fault recovery time rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. Similarly, TMA has reached 8, 13, 20, 28, 37, 46 and 57 fault recovery time rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. In addition, HAMR has achieved 9, 15, 22, 30, 40, 50 and 62 fault recovery time rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. But the developed AFT-HCARP protocol significantly outperformed as 4, 7, 10, 13, 17, 21 and 26 for various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 while comparing the traditional models.

5.2.3 Packet delay

In any network routing system, packet delay is the undesirable events occurred in the network path. Moreover, the packet delay can create the entire system unfit for existence. Moreover, packet delay is computed using eqn. (14),

$$P_d = T_p(t) - R_p(t) \quad (14)$$

Where, $T_p(t)$ is denoted as number of transmitted packets at time t and $R_p(t)$ is number of received packets at time t .

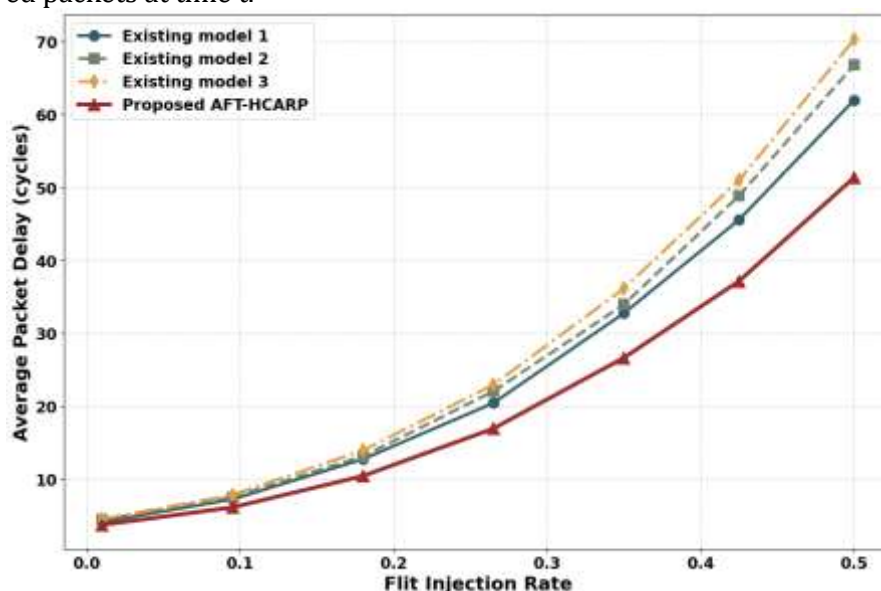


Fig.14 Comparison of average packet delay

Here, the average packet delay analysis of the developed AFT-HCARP protocol performance is compared with traditional routing algorithms such as FDNT, TMA and HAMR which are demonstrated in fig.14. Initially, FDNT protocol has attained 4.2, 7.3, 12.8, 20.5, 32.8, 45.6 and 62 average packet delay rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. Similarly, TMA has reached 4.5, 7.6, 13.2, 21.1, 34, 48.9 and 66.8 average packet delay rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. In addition, HAMR has achieved 4.6, 7.9, 14, 23, 36.2, 51 and 70.3 average packet delay rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. But the developed AFT-HCARP protocol significantly outperformed as 3.8, 6.2, 10.5, 17, 26.7, 37.2 and 51.4 for various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 while comparing the traditional models.

5.2.4 Throughput

Throughput of the developed AFT-HCARP protocol is termed as total amount of successful packets transmitted from source node to destination node. In addition, throughput is efficiency the network capability.

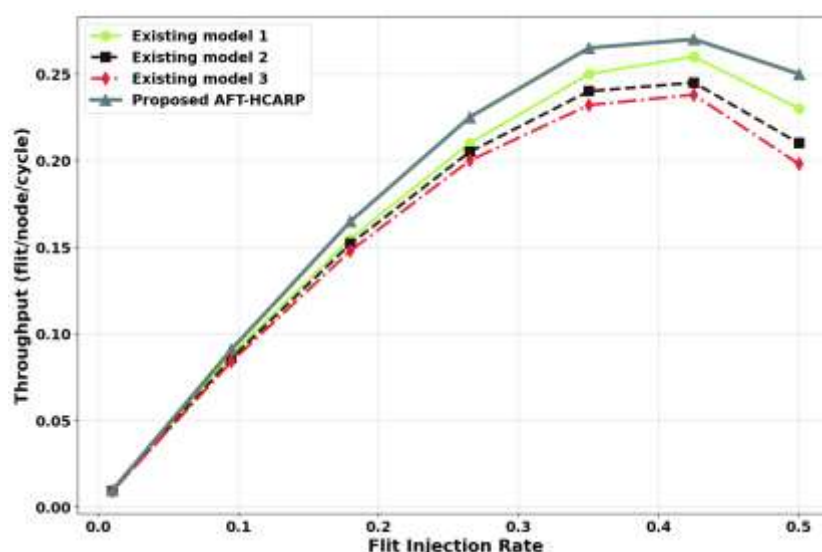


Fig. 15 Comparison of throughput

Here, the throughput analysis of the developed AFT-HCARP protocol performance is compared with traditional routing algorithms such as FDNT, TMA and HAMR, which are demonstrated in fig.15. Initially, FDNT protocol has attained 0.0095, 0.088, 0.155, 0.21, 0.25, 0.26 and 0.23 throughput rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. Similarly, TMA has reached 0.0093, 0.086, 0.152, 0.205, 0.25, 0.245 and 0.21 throughput under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. In addition, HAMR has achieved 0.0092, 0.084, 0.148, 0.2, 0.232, 0.238 and 0.198 throughput rates under the various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 respectively. But the developed AFT-HCARP protocol significantly outperformed as 0.0098, 0.091, 0.165, 0.225, 0.265, 0.27 and 0.25 for various flit injection ratio such as 0.01, 0.095, 0.18, 0.265, 0.35, 0.425 and 0.5 while comparing the traditional models.

6. Conclusion

This paper presents an effective AFT-HCARP protocol which is based on to overcome some of the issues that have driven the development of M-SoCs. The research also analyses the optimization or improvement objectives of the research papers, identifying trends that show the importance and impact of the most exploited areas and those that are becoming increasingly relevant. Additionally, the paper identifies routing algorithms such as adaptive and deterministic that implement more properties and classifies them according to the specific type of congestion level of the developed AFT-HCARP protocol. The main objective of this research is to provide the best routing performance under the M-SoCs management and the integration of AFT-HCARP protocol. Finally, the model performance like throughput (0.0098 flit/node/cycle), average packet delay (3.8 cycles), PDR (97.60%) and fault recover time (4 cycles). In future, the work will further enhanced by applying the optimization based Machine Learning (ML) parameters are used to train the hyper parameters and optimize better performance in terms routing performance and congestion control.

References:

- [1] Morell, José Ángel, and Enrique Alba. "Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices." *Future Generation Computer Systems* 133 (2022): 53-67.

- [2] Tsoukas, Vasileios, et al. "Development of a dynamically adaptable routing system for data analytics insights in logistic services." *Analytics* 2.2 (2023): 328-345.
- [3] Fang, Juan, et al. "TB-TBP: a task-based adaptive routing algorithm for network-on-chip in heterogenous CPU-GPU architectures." *The Journal of Supercomputing* 80.5 (2024): 6311-6335.
- [4] Rao, Zheheng, et al. "DAR-DRL: A dynamic adaptive routing method based on deep reinforcement learning." *Computer Communications* 228 (2024): 107983.
com
- [5] Ottaviano, Daniele. "Real-Time Virtualization of Mixed-Criticality Heterogeneous Embedded Systems for Fusion Diagnostics and Control." (2025).
- [6] Spieck, Jan, Stefan Wildermann, and Jürgen Teich. "A Scenario-Based DVFS-Aware Hybrid Application Mapping Methodology for MPSoCs." *ACM Transactions on Design Automation of Electronic Systems* 29.4 (2024): 1-43.
- [7] Lopes, Geaninne, César Marcon, and Fernando Moraes. "Application of Machine Learning Techniques for Secure Traffic in NoC-based Manycores." *arXiv preprint arXiv:2501.12034* (2025).
- [8] Dayarathna, Miyuru. "Automatic Tuning and Software Modularization Techniques for Optimized Large Scale Data Analytics."
- [9] Benabdenbi, Mounir. Contributions to the Test, Fault Tolerance and Approximate Computing of System on a Chip. Diss. Université Grenoble Alpes, 2023.
- [10] Siddagangappa, Renu. "Asynchronous NoC with fault tolerant mechanism: a comprehensive review." 2022 Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON). IEEE, 2022.
- [11] Guo, Bo, Huaming Liu, and Lei Niu. "Network-on-Chip (NoC) applications for iot-enabled chip systems: latest designs and modern applications." *International Journal of High Speed Electronics and Systems* 34.03 (2025): 2540027.
- [12] Zerdani, Amina, and Fateh Boutekkouk. "An Overview of Formal Verification of Network-on-Chip (NoC) Methods." *International Conference on Computing and Information Technology*. Cham: Springer Nature Switzerland, 2023.
- [13] Li, Zifei. Multicast router with fault tolerance mechanism for large-scale neuromorphic computing system. Diss. Technische Universität Wien, 2022.
- [14] Nagalaxmi, T., E. Sreenivasa Rao, and P. Chandrasekar. "Design and Development of Low Power Clock and Data Recovery Circuit for Asynchronous Network on Chips." *Journal of Integrated Circuits and Systems* 17.3 (2022): 1-9.
- [15] Muhsen, Yousif Raad, et al. "Routing Techniques in Network-On-Chip Based Multiprocessor-System-on-Chip for IOT: A Systematic."
- [16] Pitre, Boisy Gene. A Hyper-Embedded Architecture for Area-Constrained System-on-Chip Applications. Diss. University of Louisiana at Lafayette, 2025.
- [17] Yasrebi, Shahrouz, et al. "A fuzzy integrated congestion-aware routing algorithm for network on chip." *Frontiers of Information Technology & Electronic Engineering* 22.5 (2021): 741-755.
- [18] Arrivé, Maël. "Determinism Enhancement in Embedded Systems for Aeronautics and Space applications." (2025).
- [19] Cinque, Marcello, et al. "Evaluating virtualization for fog monitoring of real-time applications in mixed-criticality systems." *Real-Time Systems* 59.4 (2023): 534-567.
- [20] Wicaksana, Arief, et al. "Maintaining Communication Consistency During Task Migrations in Heterogeneous Reconfigurable Devices." *Multi-Processor System-on-Chip 1: Architectures* (2021): 255-285.
- [21] Shuvo, Amit Mazumder, et al. "A comprehensive survey on non-invasive fault injection attacks." *Cryptology ePrint Archive* (2023).

- [22] SB, Sujata, and Anuradha M. Sandi. "Design and Implementation of High QoS 3D-NoC using Modified Double Particle Swarm Optimization on FPGA."
- [23] Ottaviano, Daniele. "Real-Time Virtualization of Mixed-Criticality Heterogeneous Embedded Systems for Fusion Diagnostics and Control." (2025).
- [24] Kaur, Simran Preet, et al. "A survey on scheduling and mapping techniques in 3D Network-on-chip." arXiv preprint arXiv:2211.02378 (2022).
- [25] Million, Davy, et al. "Depth-First: A Deterministic and Scalable NoC Routing Protocol for 3.5 D Packaged Architectures." IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2025).
- [26] Reddy, B. Naresh Kumar, Md Zia Ur Rahman, and Aime Lay-Ekuakille. "Enhancing Reliability and Energy Efficiency in Many-Core Processors Through Fault-Tolerant Network-On-Chip." IEEE Transactions on Network and Service Management 21.5 (2024): 5049-5062.
- [27] Spieck, Jan, Stefan Wildermann, and Jürgen Teich. "A Scenario-Based DVFS-Aware Hybrid Application Mapping Methodology for MPSoCs." ACM Transactions on Design Automation of Electronic Systems 29.4 (2024): 1-43.