JOURNAL OF BASIC SCIENCE AND ENGINEERING

# A Concise Review of 2D Mesh Topology for Networks-on-Chip and Its Widely Used Routing Strategies

**Chekuri Snehith Sri Sai[1] , Mukkala Yugandhar[2] , Guduru Madhu Sudhan Reddy[3] , Aashi Sah[4] , Prof. Megha Trivedi[5]**

[1,2,3,4,5] Department of Computer Science & Engineering, Parul University, Vadodara, 391760 India.

**Keywords:**
Network – On – Chip, Routing Algorithm, Adaptive, Partially Adaptive, Hybrid routing.

https://doi.org/10.10499/JBSE.20261435342

**ABS TRAC T**

Networks – on – Chip (NoCs) have gained popularity as the most dominant interconnect paradigm for many-core and accelerator-rich Systems-on-Chip (SoCs). Among the many topologies available, the two-dimensional (2D) mesh topology serves as the canonical architectural template. This survey presents a concise review of 2D mesh–based NoC architectures.

Routing algorithms are classified into deterministic, partially adaptive, fully adaptive, and non-minimal categories. A concise survey of popular routing strategy till date has been discussed for mesh NoC topology along with the advantages and disadvantages of the routing strategy.

## 1. INTRODUCTION

The ever-growing demand for newer SoCs, and owing to the technological advancement in semiconductor research, makes it possible to place thousands and thousands of Intellectual Property (IP) cores on a single die. The effectiveness of this system relies on swift communication among the many cores and hence, the on- chip communication fabric has emerged as a prominent framework of overall system performance, energy efficiency, and reliability[1].

The earlier SoC communication relied heavily on a Bus – Interconnect or a Point – To – Point communication architecture, which is both inefficient and suffers from poor scalability. This led researchers to think in a direction where they can put a well-defined network in place. Thus, the paradigm shifted from designing complex SoCs from the scratch to placing well defined Network – On – Chip (NoC) which will work as a skeleton for placing the IP cores which form the SoCs[2], [3].

The NoC, is the much-needed approach which can outsmart the traditional choices of interconnect by replacing it with structured packet-switched networks, which have better scalability, higher communication bandwidth, and superior architectural modularity for many-core systems[4].

Among many NoC topologies, the 2D mesh and tori are heavily used for commercial HPC solutions [5] due to its regular layout, simple XY-routing deadlock freedom, and amenability to tiled floor planning[6]. However, the conventional mesh exhibits suboptimal path diversity, diameter growth proportional to $\sqrt{N}$, where $N$ is the total number of Processing Elements and vulnerability to process variation and thermal hotspots. These limitations have motivated a rich ecosystem of topological derivatives and routing enhancements. For instance, recent works have explored wire-maximal networks[7] and fault-tolerant architectures[8] to address these scalability concerns.

This paper reviews the evolution of interconnection technologies and network topologies for **2D Networks-on-Chip (NoCs)** with special emphasis on Mesh and its popular routing strategies over the past decades, focusing on the lack of an integrated perspective between IC interconnects and NoC architectures. By



Vol. 23, No. 1, (2026)
ISSN: 1005-0930

characterizing 2D NoC Mesh topology based on theirrouting decisions, this paper highlights their impact on

communication performance, scalability, and efficiency. As modern computing systems demand increasingly efficient designs, understanding the limitations of the existing routing algorithms and emerging trends enables the co-optimization of interconnect technologies and network topologies for future applications.

### 1.1 Definitions and Terminology

Key definitions utilized throughout this report include:

- **Network-on-Chip (NoC):** A packet-switched on-chip communication framework which comprises routers or switches, network interfaces (NIs), and links interconnecting the components.
- **Topology:** The graph $G(V,E)$ defining network nodes ($V$) and communication channels ($E$).
- **Routing Algorithm:** It is function that determines the packet paths. Routing algorithms are often classified as deterministic, adaptive, minimal or non-minimal[9].
- **Virtual Channel (VC):** A logical lane multiplexed over a physical channel[2].
- **Bisection Bandwidth ($B_b$):** Aggregate bandwidth across the minimum cut dividing the network into two equal halves.
- **Deadlock Freedom:** Absence of cyclic dependencies on channel resources, often formalized via channel dependency graphs (CDG)[10], [11].

### 1.2 Taxonomy and Nomenclature

A structured taxonomy of mesh-based topologies is essential for understanding the design space. The classification is organized based on three orthogonal axes: link augmentation[12], [13], [14], [15], node aggregation[16], [17], [18], and dimension transformation[19]. The figure 1 show the classification tree, However, this paper focuses on the Mesh topology and its popular routing strategies.

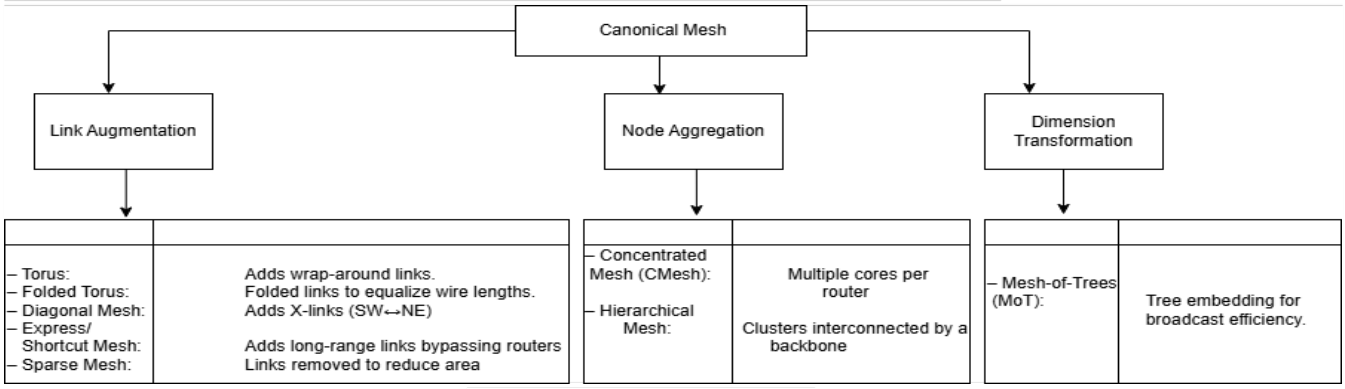



**Figure 1: Classification tree**

## 2. Mesh in a historical perspective

Clark D. Thompson [20] evaluated mesh-type computer interconnection networks and derived lower and upper bounds on the time required to broadcast a message. Primarily Thompson's work explicitly discussed Generalized Connection Networks (GCN) for parallel processor intercommunication, it also evaluated Single Instruction Multiple Data (SIMD) computers and mesh-type arrays in the context of message routing algorithms. In particular, Thompson specifically considers a *square mesh-type array* when discussing the routing steps required for executing an algorithm on an *N*element computer.

Mesh connected computers (MCC)[21] is the choice of architecture for low image processing tasks because of its strong local communication prowess i.e. two neighboring pixels may be mapped to adjacent

processing elements directly connected via a communication link. The work by Miller and Stout in [21], demonstrated that, the MCC is equally powerful and efficient in processing high – level image operations that involve geometric properties. The work proposed and evaluated a well-known algorithm for MCC for extreme points and the convex hull of individual image component calculation, connectivity assessment of the image components, inter component distance calculation, bilevel median filter operations etc. and achieved an optimal time complexity of O($n$), for a mesh of $n \times n$ where $n = \sqrt{N}$, which is a significant improvement from previous O($n^2$). This demonstrates the notable speedup offered by the mesh architecture.

Siegel at [22] went on to define an interconnection network which is a set of interconnection function, where each function defines a bijection over the set of processing element (PE) addresses. Whenever an interconnection function is applied, it transfers the content of one PE and sends it to the PE for which the data is destined. The work in [22] evaluates five different interconnections including the Illiac IV system of [23]. The work described the Illiac IV interconnection to be of N PEs, arranges in a perfect square having each side $\sqrt{N}$, each PE connects to its north, south, east, and west neighbors.

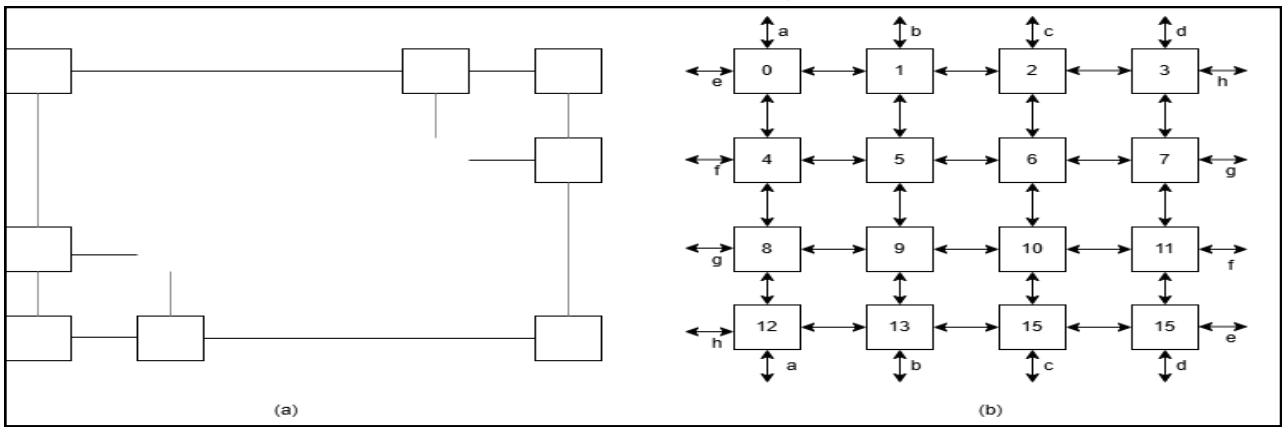



**Figure 2: The interconnection (a) A mesh connected computer[20] (b) The Illiac IV[22]**

Further, a d – dimensional computer is described by[24], which is a distributed memory parallel computer composed of processing elements. These elements are connected as a d – dimensional cube, essentially the mesh, where the PEs are connected to its neighbours, two in each dimension, provided they exist. Further, the authors went on to describe four mesh connected SMID computers namely Cellular Logic Image Processor (CLIP4) developed for specific image processing tasks, Geometric Arithmetic Parallel Processor(GAPP) often attributed as the first mesh connected image processing computer adapted commercially and apart from image processing it is also efficient for pattern recognition computations, Massively Parallel Processor(MPP) which is implemented with High-speed Complementary Metal-Oxide- Semiconductor (HCMOS) using 5 micron design rules having 16K bit-serial arranged in a square mesh of 128 x 128 PEs and operates under uniform addressing communication scheme governed by a specifically designed controller, and finally The MasPar MP-l commercial machine having 16K processors and the communication is based on X – Net configuration which enables a PE to connect to 8 neighbouring PEs.

The predominant emphasis on 2D mesh architectures underscores their practicality and ease of implementation, as compared to higher-dimensional or more complex network topologies available during that period. Scalability of mesh networks and its appropriateness for SMID computers made it a very strong candidate for the interconnection network.

### 2.1 Mesh as On – Chip Network

Traditionally, SoCs provided no flexibility in the interconnectivity of computational, storage, and I/O resources, these are generally fixed at design time itself. This rigidity became a limiting factor in terms of

reusability, specially when the demand shifted towards more software-oriented approach which promotes post designmodification.Though,Field-programmablegatearrays(FPGAs)provideflexibleinterconnect which arereprogrammableaswell,thefinegranularityintheircomputationalresourcesstressesoutthe interconnection fabric and becomes a scalability concern. The new breed of FPGA has well adapted towards the softwarecentricdesignofcomponentsbyprovidingembeddedprocessingcoressuchasIBM'sPowerPC, ARC cores,andMotorolacores.ArchitecturessuchasChameleon,whichintegrateanembeddedARCcore with a reconfigurablecomputingfabric,provideinspirationforthedesignoffutureNetwork-on-Chip(NoC) platforms catering the billion-transistor era [25].

Jantschin[26]showedaNoCinitssimplestformasamesh,wheretheinterconnectionbetweenthe resources (processors, DSPs, I/O peripherals etc.) are through the switches of the network and it is largely independent of the size of the network. It outlined that, in the mesh routing of messages is easy and therefore switches employed to carry out the task is smaller in size, have larger bandwidth and are scalable.

Furthering the work, Shashi Kumar et al. in proposed a packet – switched platform on a single chip that scaleswellforanynumberofprocessingelements,theyreferredthisplatformasNetwork–On–Chip (NoC). NoCencompassesboth,thearchitectureandthedesignmethodology.EssentiallytheNoCarchitecture is the backboneforallthecommunicationrequirementsoftheOn–Chipresources.NoChasamodular design andcanbeputtogethertoformalargerOn–chipnetworkjustbyconnectingthebuildingblocks which otherwisearestandalonepieceofhardware,also,itsisverymuchadaptabletopracticalcommunication scenarios. For the said network mesh topology have been selected owing to its advantages as simple realization and the fact that the underlaying network offers local interconnection between the resources and switches which are independent of the size of the network. The switches and resources are connected via channels in a Manhattan-like structure which offers seamless communication between various components. An internalswitchinthesaidnetworkisconnectedtofourotherswitchesinup,down,leftandrightdirection, however the corner switches and edge switches are connected to two and three other switches respectively. The switchesarecapableofsendingmessagesonehopatatimetothedestinationviaswitcheswhichare connected to broadcasting switch. The internal congestion is handled via internal queues and in some cases buffers. ThisapproachisnamedasCLICHÉ(Chip-LevelIntegrationofCommunicatingHeterogeneous Elements).

### 2.3 Mathematicaldefinitionofmesh

A two-dimensionalmesh $M(n,m)$ isdefinedasadirectedgraph $G(V,E)$ thesetofvertices $V$ isgivenby

$$V = \{v_{ij} | 0 \le i < n, 0 \le j < m\} \quad (1)$$

Essentially characterize an $n \times m$ nodes arranged in a grid fashion where each node is addressed by its subscripted $i$ and $j$ indices whichcanbeperceivedasrowandcolumnindicesrespectively.

The edgeset $E$ consistsofdirectedlinksbetweennodesthatareaxisalignneighborsinthegrid.More specifically

$$E = \{(v_{i,j}, v_{i+1,j}), (v_{i+1,j}, v_{ij}) | 0 \le i < n-1, 0 \le j < m\}$$
$$\cup \{(v_{i,j}, v_{i,j+1}), (v_{i,j+1}, v_{i,j}) | 0 \le i < n, 0 \le j < m-1\} \quad (2)$$

Thus, each node $v_{i,j}$ is connected by directed edges to its north, south, east, and west neighbors. Note that, the definition above aligns with the definition given by[27].

## 3. Popular Routing in Mesh

Here we present the popular routing proposed by various researchers for the Mesh NoC topology. We

reproduced some of the popular routing strategies which becomes baseline for the rest of the routing algorithms. Whenever possible, extensive figures illustrated the routing strategy derived from the baseline routing algorithms.

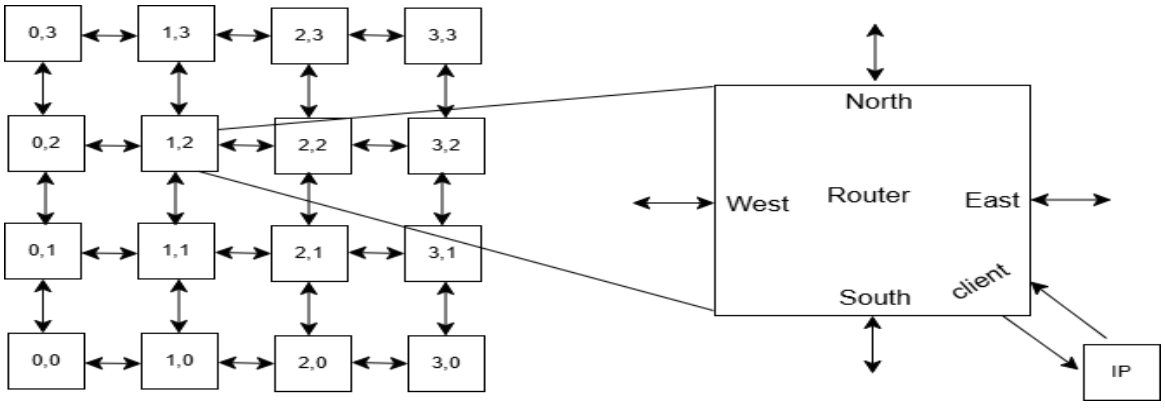
Routing algorithms[28] are most often categorized as deterministic routing and adaptive routing. Deterministic routing happens through a fixed path from start to finish that doesn't change based on the conditions of the network (i.e. overloaded, down). Deterministic routing can be Characterized as an established path exists for any source—destination, x, y coordinates, there are no on-the-fly, mid-travel decisions based on traffic, simple to make a router perform this function and minimal hardware requirements, delay andorderingarepredetermined.Thekeydisadvantagesofthedeterministicroutingare,itsuffersfrom poor loadbalancingandthenetworksaturatesquicklywithhighinjectionrate.

On theotherhand,AdaptiveRoutingprovidesdifferentpathsbasedonthenetworkstatusatagivenpoint in time;forinstance,linkusage,linkcongestion,availablebuffers,failedlinks,etc.itcanbeformally characterized by the exploration of multiple paths between source to destination, dynamically making routing decisions, logging of network state information, necessitates more control logic, these class of algorithm may further classifiedas **partially adaptive routing** whichpermitspathdiversityonlywithinalimitedsubsetof routes, turnmodels[29]areoneoftheexamplesand **Fully adaptive routing** whichallowsanyminimal(and sometimes non-minimal) routing path. Advantages include improved load balancing, Improved throughput in high congestionsituations,Improvedfaulttolerance.However,complicatedrouterarchitecture,elevated power/area cost and less predictable latency remain its greatest disadvantages.

It is wormholeswitchingthatdrivesextensiveuseofdeterministicrouting.Wormholeswitchingneeds very smallbuffers,meaningsmallandfastrouters.Ifonestageissignificantlyslowerthantherest,pipelined operation is of no use. Thus, wormhole routers typically use the routing algorithm in hardware[28].

The simplest (deterministic) routing policy first addresses the coordinate displacement of one dimension and subsequently,thedisplacementoftheotherdimension.Thistypeofpolicyiscalled**dimension-order routing**, which traverses packets along network dimensions with a fixed static order and completely resolves deviation in one dimension before proceeding to the next[28].

The authors at [30] have presented an analysis of XY routing for Touchstone Delta Concurrent File System way backin1993,itstatesthat,physically,flits(flowcontroldigits)traversethenetwork.Packetsarerouted according to the XY routing protocol, which means that the direction of routing is contained in the message header. FortheTouchstoneDeltanetwork,thismeansthatitisfirstroutedacrosstheXdimensionandthen routed downtheYdimensiononcethecorrectcolumnisreached.Thisalgorithmisessentiallyaclassof dimension – order routing as per [28]. The XY coordinated mesh topology and a higher level abstraction of its router with its attached IP is shown below in figure 3.






Vol. 23, No. 1, (2026)
ISSN: 1005-0930

**Figure3: A 4 × 4 mesh topology showing XY coordinates with its router structure[31]**

We give the XY routing algorithm as presented in[28], [31] which is a class of dimension order routing (DOR), here for a ready reference.

Algorithm1: XY Routing Algorithm

```
Input
    Current node coordinates {x_s,y_s}
    Destination coordinates {x_d,y_d}
Output
    nextoutput port
Begin
    do
        If(x_s< x_d)
            Go East
        else if (x_s> x_d)
            Go West
        else
            required x coordinate achieved
    while(x_s ≠ x_d)
    do
        If y_s < y_d
            Go North
        else if (y_s>y_d)
            Go South
        else
            required y coordinate achieved and hence destination
    while (y_s ≠ y_d )
End
```

While XY routing ensures deadlock-free operation, it is not an adaptive routing scheme[32]. Although authors at [31] disagree with the freedom from deadlock claim of XY routing, however one can always use virtual channels to escape from deadlock[33].

Randomized, Oblivious, Multi – phase, Minimal (ROMM)[34], was introduced to eliminate some of the shortcomings of DOR such as degraded worst-case behavior, poor load balancing and unacceptable latency at high network load scenarios. ROMM is a randomized oblivious routing strategy which always chooses the minimal path from the source to destination in a phase wise manner. ROMM selects $n-1$ number of randomly selected nodes between the source and destination pair along its minimal path. Thereafter, the packet is forwarded to the destination through these randomly selected nodes in a $n$ phase ROMM. In each phase, the routing takes a dimension order routing to progressively reach to these intermediate nodes and finally at the last phase the packet is delivered to the destination. As already mentioned, DOR is a deadlock and live lock free routing strategy and therefore ROMM is as well. Whenever, the number of phases $n$ increases, the randomness of the algorithm increases as well, however, the cost of routing, in terms of network resources, such as, memory size, logic complexity and the maintenance $n-1$ intermediate nodes.

Deterministic routing algorithms, such as XY, and ROMM have moderate latency when network traffic is

relatively small. However, as network congestion increases, the networks performance drops[35]. This lead researchers to speculate in the direction of partially adaptive and adaptive routing strategies.

The Turn Model of Glass et al [29] is a deadlock-free partially adaptive routing protocols for interconnection networks, since it assesses routing as establishing a channel (communication path/channel edge) and uses the number of turns (number of node paths) to dictate deadlock prevention. The Turn Model takes advantage of deadlock freedom by not allowing the minimum number of turns needed to avoid a cyclical dependency created by using channel in/out and allows for adaptive routing freedom without the cost of additional turns. Compared to dimension-ordered routing which can take additional turns through ordered access and thus jeopardizes deadlock freedom more by giving it an extra turn, the Turn Model assumes turns will be less through a more flexible approach and access through ordered dimensions provides too much ease for deadlock prevention but not enough turns. The Turn Model creates West-First, North-Last, and Negative-First routings which don't necessarily prevent access, but still adaptable down other paths. Additionally, the Turn Model does not require virtual channels to maintain deadlock freedom. This low-cost implementation of restrictions on certain sections of the network avoids additional complexity for the router and increased hardware overhead costs, which is why the Turn Model is commonly used and is still applicable in mesh-based Networks-on-Chip and fault-tolerant routing.

TurnModelsrendersanunevenadaptiveness,thatis,abouthalfofthesource–destinationpairsarerestricted toasingleminimalpath,whiletherestofthenodesretainfulladaptiveness.Resultinginunfairpathutilization andreducedabilitytotacklecongestion,whichagainbecomesacontributingfactorindegradingthenetwork performance. Similarly, XY routing achieves deadlock freedom by prohibiting specific turns[32].

TheOdd–Even(OE)routingalgorithm,presentedbyChiu[32],isadeadlock-freepartiallyadaptive, routingalgorithmfor2Dmeshnetworks.ThealgorithmisbasedontheTurnModelsproposedbyGlasset al[29].Unlikeitspredecessor,thatapplyuniformturnrestrictionsacrossthenetwork,OEroutingapplies **column-parity–dependent turn restrictions**, resulting in a fair amount of adaptiveness.

InOEroutingstrategy[29],meshcolumnsaredesignatedas **odd** or **even** .Anymeshnodegivenbyits coordinates $(x,y)$ seeFigure3iscalledevenifthevalueofthe $x$ coordinateiseven,otherwiseitisdesignated as odd.Turnrestrictionsareappliedselectivelybasedonthisparity.Specificturns,forexampleEast–North and East–Southturnsarenotallowedforanypacketcurrentlyinevencolumn,North-WestturnsandSouth – Westturnsarenotallowedforapackettotakeifitiscurrentlyinanyoddcolumn.Theseturnsarerestricted to eliminatecyclicdependenciestoavoiddeadlock.EssentiallyOEroutingalgorithmandturnmodelsare developedtoavoidvirtualchannelswhichcomplicatestherouterdesign.WenowreproducetheOErouting given at [32] below as algorithm 2.

Algorithm 2: Odd – Even turn model

Input: Source $(x_s,y_s)$, Destination $(x_d,y_d)$, Current$(c_x,c_y)$
Output: Set of Allowed direction ports $AlloweddirectionPort$
Begin
    $AlloweddirectionPort \leftarrow \emptyset$
    $xc \leftarrow x_d - c_x$
    $yc \leftarrow y_d - c_y$
    if ($x_c$=0 ∧ $y_c$ =0)
        *//already at destination node*
        Deliver the packet to the Local and exit;
    if ($x_c$=0)
        if ($y_c$> 0)
            $AlloweddirectionPort \leftarrow \{North\}$
        else
            $AlloweddirectionPort \leftarrow \{South\}$
    else
        if ($x_c$>0)  *//east routing*



Vol. 23, No. 1, (2026)
ISSN: 1005-0930

```
        if (y_c = 0)
            AlloweddirectionPort ←{East}
        else {
            if (c_x%2 ≠0 ∨c_x= x_s)
                if(y_c>0)
                    AlloweddirectionPort ←{North}
                else
                    AlloweddirectionPort ←{South}
            if (x_d%2≠0 ∨x_c≠1)        //destinationatoddcolumn or 2 hops away
                AlloweddirectionPort ←{East}
            }
    else {          //westrouting
        AlloweddirectionPort ←{West}
        if (c_x%2 = 0)
            if (y_c>0)
                AlloweddirectionPort ←{North}
            else
                AlloweddirectionPort ←{South}
        }
    Selecta portfromtheset AlloweddirectionPort to forward the packet
End
```

DyAD or Dynamic Adaptive Deterministic switching[36] was proposed utilizing the wormhole switching andhas been tested for 2d mesh NoC. The authors combined two routing strategies for this purpose and has given a router design to carry out the task. The proposed routing scheme, adapts to the network's current congestion. Each router continuously monitors its local traffic and adjusts its routing decisions accordingly. When the network is not congested, DyAD routers use a deterministic mode, XY routing strategy in this case, which reduces routing delays, a key feature of deterministic routing. When congestion is detected, the router switches to an adaptive routing mode i.e Odd – Even routing strategy in this case, allowing traffic to be steered away from congested links by exploiting alternative paths. This dynamic adaptation results in improved network throughput, which is especially useful for applications implemented using the Network-on-Chip (NoC) paradigm. We note that, both XY(deterministic) routing scheme and Odd – even(adaptive) routing scheme are deadlock free and therefore, the DyAD also avoid deadlocks and livelocks.

Dynamic XY (DyXY) routing[37], a congestion-aware adaptive routing algorithm, facilitates adaptive routing predicated on localized congestion metrics, all the while guaranteeing both deadlock and livelock freedom. The localized congestion is determined by a factor which is dependent on the queue length of the router at the moment of routing in a particular direction, the authors termed it as *"stress value"*. Needless to say, the packets in DyXY algorithms shall always be forwarded to the router towards the destination having the lowest stress value. The adaptability inherent in DyXY is derived from its capacity to modify routing choices contingent upon the observed congestion levels surrounding each router. Simultaneously, the algorithm's deadlock and livelock-free characteristics are ensured by confining packet always take the shortest pathsbetween the source and destination pair. For realization of this algorithm a router microarchitecture was alsoproposed by the authors having a set of FIFO input buffers, stress value counters, four in number, one foreach direction, a input arbiter, A history buffer serves to log channels that have sent input requests, a controller and a crossbar fabric. We now resent the algorithm DyXY below.

Algorithm 3: DyXY
Input: Current_Location$(x_c, y_c)$ , Destination$(x_d, y_d)$ , Stress_Neighbour$(E,W,N,S)$

```
Output: Output_direction
Begin
while(Not done){
    if(x_c = x_d ∧ y_c = y_d)
        destination router reached,deliver packet and exit

    else{ dx= x_d − x_c
          dy=y_d−y_c
          Output_direction ←∅
          X_direction←∅
          Y_direction ←∅
                if(d_x≠0)
                    X_direction←{E,W,Stress_Neighbour(E,W,∞,∞)}
                if(d_y≠0)
                        Y_direction←{N,S,Stress_Neighbour(∞,∞,N,S)}
          Output_direction←min (X_direction,Y_direction)
          forward packet to Output_direction
          }
    }
End
```

Since,intheDyXYalgorithmaspresentedabove,apacketcanmovein both the directions X and Y dynamicallydependinguponthestressvaluerepresentingthecongestionscenario, DyXY algorithm uses virtualchannels(VC)i.e.twoVCsalongtheYaxisandoneVCalongtheX axis[38]. An improvement of DyXYisproposedin[39],calledEDXY,herethecongestioninformation is propagated using a 2 bit congestionflaginX–directionfortheportsinEastandwestdirectionand Y – direction for the ports in SouthandNorthdirection.Ateachhop,thecongestioninformationispassed to all the successive node in a particulardirection,wheneveranodedeterminesitsinputqueueisoccupied beyond a certain threshold, it propagatesitsowncongestioninformationtothesuccessivenodeinthedirection of its output port(X or Y direction) along with the congestion information of its predecessor node.

Anotherrouting,calledtheDeflectionrouting[40]worksasadecentralized and adaptive routing method, particularlysuitedforNetworks-on-Chip.Adeflectionswitchdiffersfromother NoC switches, as it doesn't haveanybufferqueues.Asaresult,packetsarealwaysintransit,movingfrom switch to switch on a cycle-by-cyclebasistowardtheirintendeddestinations.Contentionforlinksaresolved by misrouting the lower prioritypacketstowardsaunfavouredlinkaccordingtoadeflectionpolicy.Three broadly available deflection policiesare,(1)Randomi.e.packetshaveequalprioritiesanduponcontention of links, deflection may take placerandomly.(2)Straightthrough,havingprioritiesofapackettravellingin straight line is higher than the packetshavingturnsandshallbedeflecteduponexperiencinglinkcontention, and (3) Weighted priority, whereapacketspriorityiscalculatedbasedontheageofthepacketinthenetwork, distance measure i.e. the distancebetweenthecurrentandthedestinationrouter,deflectiontimes,which records the number of time thisverypackethavebeendeflectedinitslifetimeandadefaultpriorityvalue assigned to the packet. The weighted priority is calculated as

$$P\omega = \omega_A \times Age + \omega_d \times Distance + \omega_{df} \times Deflectiontimes + \omega_p \times Priority\ Initial \quad (3)$$

Whereis$\omega$theweightassociatedwitheachofthecomponent.Apartfromthese, two more network factors havebeendefinedin[40]whichare,which*Deflection index* is count induced due to single deflectionintheshortestpathofapacketandwhen*Packet travel ratio* is more than one shortest

path between the current and the destination router. Note that, when a packet is marked as don't care, it can be routed via any of the shortest path without deflection, of course the path chosen must be free of contention. The flits or packets are always on a run i.e. a flit will be allocated an output channel invariably irrespective of its destination (towards destination or deflected due to congestion essentially misrouted) and therefore no deadlock is possible. Age factor in assigning priority eliminates lovelocks as well. Also, a buffered deflection routing is proposed in [41], where, a central buffer of depth $Nb$ is placed in each router is placed, these buffers are utilized by all the ports of a router and hence it is called a central buffer pool. The routing deflection take the same philosophy as buffer less deflection with the difference that, packets shall be deflected only when the buffer pool is critically occupied. These buffers are divided into D groups such that each group has $\frac{N_b}{D}$ buffers. Each group is given to a port. These groups are maintained in a ring fashion, where shifting of the group is done every half cycle in a clockwise direction.

Another XY based adaptive routing strategy which uses context-aware agents was proposed in [35] for the mesh NoC topology, here the routers are name as agents which collaborate with its neighbours to assess the congestion of the network, by learning from the experiences of the neighbouring routers, the network here is named as society. This algorithm also behaves in a hybrid manner i.e. deterministic when network load is moderate, but switches dynamically adaptive routing whenever the network load rises beyond a threshold. The collaboration is done using 2 bit values in each ports of the router(North port, south port, east port and west port). These bits show the quantized value of the load or the fullness or emptiness of the internal buffers i.e. 00 signifies port is free, or the buffers are empty, 01 implies the buffers are 50% full, 10 indicates the buffer is 75% full and a value 11 signifies the port is unavailable due to heavy traffic condition. This algorithm also exhibits fault tolerant nature, in the event of a malfunction, the packets are rerouted using the knowledge of the neighbouring agents to isolate the malfunctioning router and reroute the packet. These agents employ reinforced learning to be aware of the context of the neighbouring area. This routing also exhibits promising results in easing the congestion hotspots and distributing it to the entire network, also it shows improvement in latency pertaining to multimedia applications.

Neighbors-On-Path (NoP) [42] another adaptive routing strategy, which provides a output port selection on the basis of not only the node ahead in the routing path, rather, it collects congestion information of the nodes in the neighbourhood i.e. at least two hops from the current node towards the destination (considering both X and Y axis adaptively) and selects a direction which is more viable in terms congestion in the path of the packet. This is unlike the DyXY or Local adaptive routing which only takes into account the immediate next hop in either direction(X or Y), towards the viable source – destination pair routing.

RCA[43] is another adaptive routing algorithm for 2D mesh NoC which essentially is a combination of scalable, lightweight methods that use congestion data from different network points to help choose ports. It utilizes a low-bandwidth monitoring network to share congestion information between neighbouring routers. Each router combines its own congestion observations with data from its neighbours. This combined congestion estimate is then used to guide port preselection and is sent upstream. During this process, the relevance of contention information is determined based on how far away it is from the current router. This reduces the impact of outdated information and prevents inaccuracies caused by non-optimal paths. Three RCA approaches were proposed, in 1D RCA, Weighted congestion data is gathered along the packet transmission path i.e. in the same dimension of the packet flow, and the resulting congestion signals are conveyed in the opposite direction as the packet flow. This data is then used by routers farther upstream to help them select egress ports based on the congestion conditions detected downstream. RCA fanin, on the other hand logs congestion information from same dimension of the packet flow along with its orthogonal direction thereby providing more regional information, however, this information may also contain

congestion values of nodes in another routing quadrantswhich may never be visited by the packet during its traversal from source to destination. Combining information from mutually exclusive routing regions in RCA Fanin minimizes noise and maximizes coverage, compared to RCA 1D.. A routing quadrant is described as thesubnetwork created by a packet's current location in the mesh and the exit port it uses to leave the current node, proceeding toward the destination port[44].

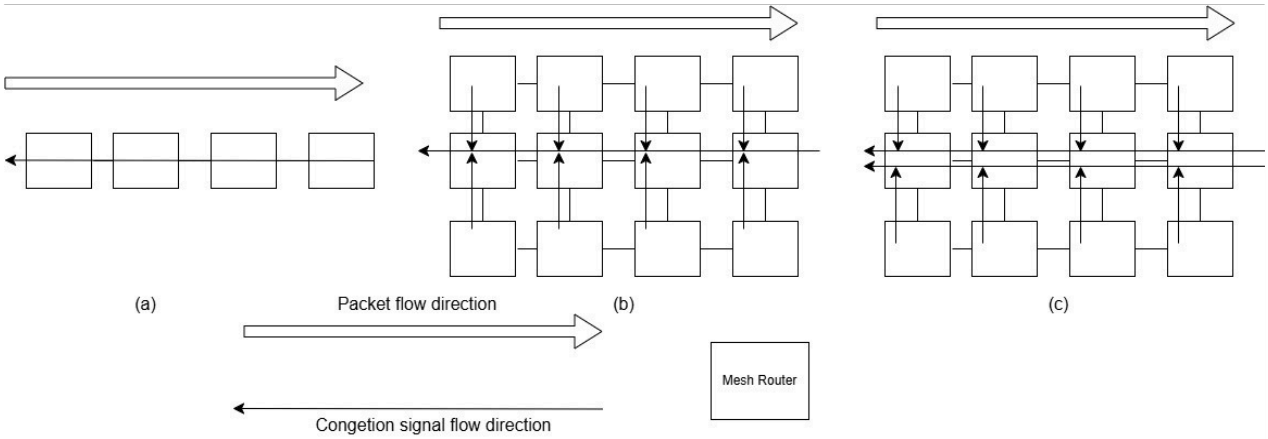
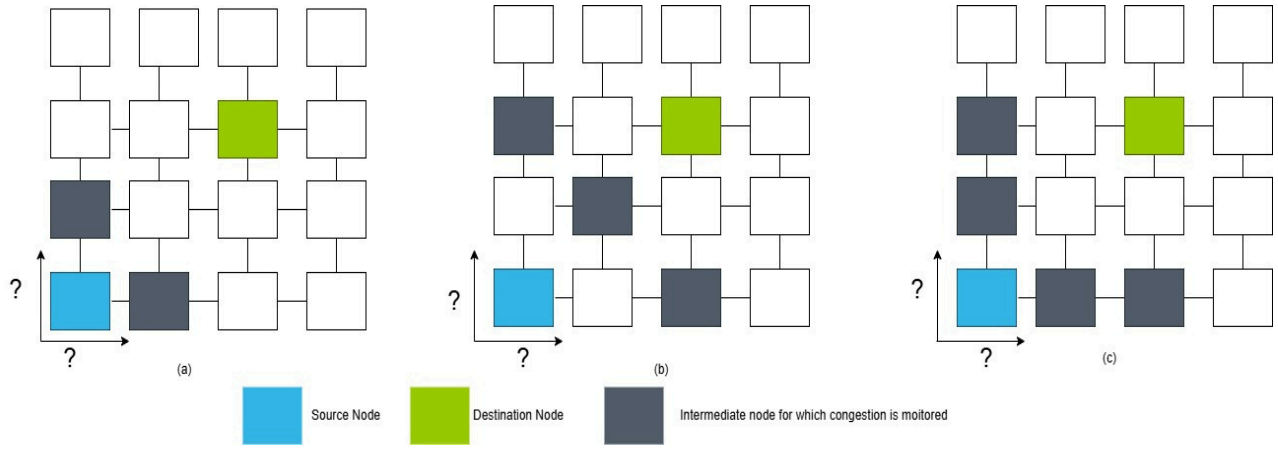



**Figure 4: RCA routing (a) 1D RCA (b) RCA Fanin (c) RCA Quadrant**

The RCA Quadrant method optimizes congestion estimations by maintaining different set of values to each network quadrant. Thereby separating the non – essential congestion values in port selection and minimizing the noise and maximizing the regional knowledge of congestion. Figure 4 illustrate this phenomenon, note that, the messages in figure 4(b) are at present somewhere in the middle row of the mesh, coursing through the input and output ports of the corresponding routers, the congestion values are propagated in the opposite direction of the packet flow. Figure 4 (c) depicts, how separate congestion values are taken from each quadrant and are propagated through the corresponding ports.

Destination-Based Adaptive Routing (DBAR) [45] take 1D RCA as its baseline and is designed to minimize the interference of non-local congestion information from non – overlapping quadrants in making thepath selection of the packet. This algorithm logs the information of the nodes which falls in the minimum quadrant defined by the current and the destination node. In order to avoid confusion between DyXY (Local), NoPand DBAR a diagrammatic comparison of both is presented below.






Vol. 23, No. 1, (2026)
ISSN: 1005-0930

**Figure 5: Congestion information recorded by different adaptive routing strategies (a) DyXY, (b) NoP (c) DBAR**

Deterministic – Adaptive Hybrid Routing (DAHR) proposed by Ji. N and Yang. Y in [46] to improve the performance of 2D mesh NoC. The baseline of the routing is the deterministic XY routing itself. Here routing is performed based on the source – destination position relationship. At the injection time, fields like hop count between source – destination pair, routing direction (RD) represented by two bits are already precalculated and loaded into the head flit of the message (wormhole switching). In a two-dimensional mesh network, RD indicates the routing direction, which is defined based on the packet's current position. A value of 00 indicates the destination is in the positive direction of X and Y axis, a value of 01 indicates the destination is in the positive direction of X and in negative direction of Y axis, 10 indicates the destination is in the negative direction of X axis and positive direction of Y axis and finally 11 indicates that the destination is in the negative direction of both X and Y axis. Off course, the value of RD is defined by the current position of the packet and the source – destination position relationship. The algorithm, first deterministically records the minimal path between the source and destination along with the RD and hop count between source – destination pair into the head flit at the injection time of the packet itself. During the routing process, the X – axis and Y – axis traversal depends upon the value of the virtual channel (vc) in the forward direction. Whenever, the count of vc does not support forwarding a packet in that direction (X or Y), the packet dynamically chooses the orthogonal direction towards the destination according to the RD value. Deadlock is avoided by allowing few turns in the quadrant marked by the RD and restricting the rest. This is shown in the figure below.

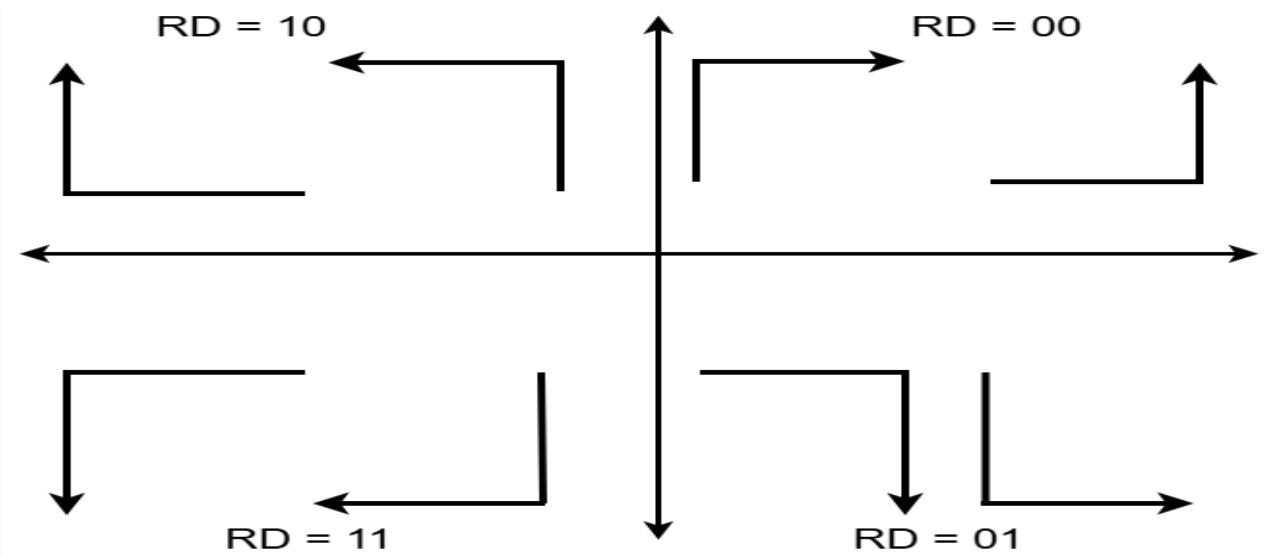



Figure 6: the quadrant marked by RD considering packet is at position 00 along with the allowed directions.

ParRouting proposed in [47] is a congestion-conscious, load-balanced routing algorithm for 2D NoC mesh, cantered on two principal foundations. Firstly, it categorizes network nodes into three priority regions according to their centrality within the network and implements specific routing strategies for each region. Secondly, it utilizes remote congestion data to inform the choice of output port, facilitating a more equitable distribution of traffic. The categorizing or partitioning of the network is done based on the centrality of the node to other nodes in the network which it can reach via a shortest distance routing, therefore, it suffices to say centrality is a function of closeness of a node to any other node in the shortest path routing between any source – destination pair. Since, in a 2D mesh NoC, different routers i.e. corner, edge and middle routers are

connected to different number of neighbours, the difference is close centrality is very stark for different nodes in the network. Weights are assigned accordingly to thelinks in the mesh NoC, for example a weight of 1 is assigned to links connecting the immediate neighboursi.e. the nodes which can be reached by a single hop from a given node. The closeness function between a pair$c$ and $d$ where $c$ is the source and $d$ is the destination is given by

$$Closeness(d) = \frac{n-1}{\sum_{c-1}^{n-1} Distance(c,d)} \quad (4)$$

Where $n-1$ is number of nodes that can reach $d$ and the factor $\sum^{n-1}_{c-1}(c,d)$ is the average distance of all shortest path to $d$. A network map is created by this closeness value obtained from equation (4), needless to say each node has its own closeness value and they vary in different segments of the network, also, it is found from the values that greater the closeness value higher is the closeness centrality of the node. It is seen that nodes in the edges have lesser closeness centrality and the nodes in the middle of the network is higher in closeness centrality, all other nodes show moderate closeness centrality. Based on this, the network priority isdistributed according to two threshold values $\tau_1$ $and \tau_2$ given by

$$\tau_1 = \kappa_1[\max(Closeness(d)) - \min(Closeness(d))] - \min(Closeness(d)), 0<\kappa_1<1 \quad (5)$$

$$\tau 2 = \kappa_2[\max(Closeness(d)) - \min(Closeness(d))] - \min(Closeness(d)), 0<\kappa_2<1 \quad (6)$$

Whenever, the closeness centrality of a node calculated using eq (4) is greater than $\tau_2$ the node, it is labelled as low priority, nodes are labelled high priority when its closeness centrality is less than $\tau_1$ are given high priority and the nodes with the closeness parameter is between $\tau_1 and\ \tau_2$ are given a medium priority. Thus, the network is partitioned into three types. The factor $\kappa_1$ *and* $\kappa_2$ plays a vital role in calculating the value of the thresholds i.e. When $\kappa_2$ is large, fewer nodes are given low priority. This shrinks the size of the central area, limiting the efficacy of routing strategies based on remote congestion. In contrast, a low $\kappa_2$ number expands the central area, i.e. lesser low priority nodes, lowering the efficiency of the priority-based routing approach in the edge region. Conversely if $\kappa_1$ is excessively large or small, it reduces the differentiation between high and medium priority nodes and does not significantly enhance load balancing performance compared to alternative routing algorithms. The idea here is to assign low priority to nodes having higher closeness centrality as it is more likely for the traffic to use these nodes in their path to their respective destinations skewing the load towards the centre. Nodes having less closeness centrality are given high priority. This is to balance the network load in an equitable manner throughout the network.

ParRouting is an adaptive, congestion aware and load balancing algorithm, it has separate algorithm for edge nodes having high priority and central nodes having low priority.

Priority-based routing for edge nodes: For nodes in edge area, first, the two prospective output nodes' priorities are compared, followed by a check of their available virtual channels (VCs). For same priority of candidate node two conditions may arise

1. Candidate nodes having same priority: If both candidates possess free VCs, or neither does, a random selection is made. However, if only one node has a free VC, that node is the one picked.
2. Candidate nodes have different priorities: If both candidates possess or lack free VCs, or if only the higher-priority node has a free VC, the higher-priority node is chosen. Conversely, if only the lower-priority node has a free VC, the lower-priority node is selected.

Remote-congestion-based routing for central node: If the source node is located in the central region, a two-stage routing strategy is employed. Initially, the candidate with a greater number of available virtual channels (VCs) is selected. If both candidates possess an equal number of free VCs, remote congestion information is assessed. Should the weighted congestion values differ, the less congested candidate is chosen;

otherwise, the output node is selected randomly. Thecongestion information is tagged along the header flit ofthe message and no separate congestion propagationnetwork has been in place. The congestion information ofthe upstream node is kept in the congestion registerofthe node and is updated whenever a flit comes from thatdirection. Also the cyclic dependencies are brokenby the use of virtual channels. The priority regions areshown in the figure below:

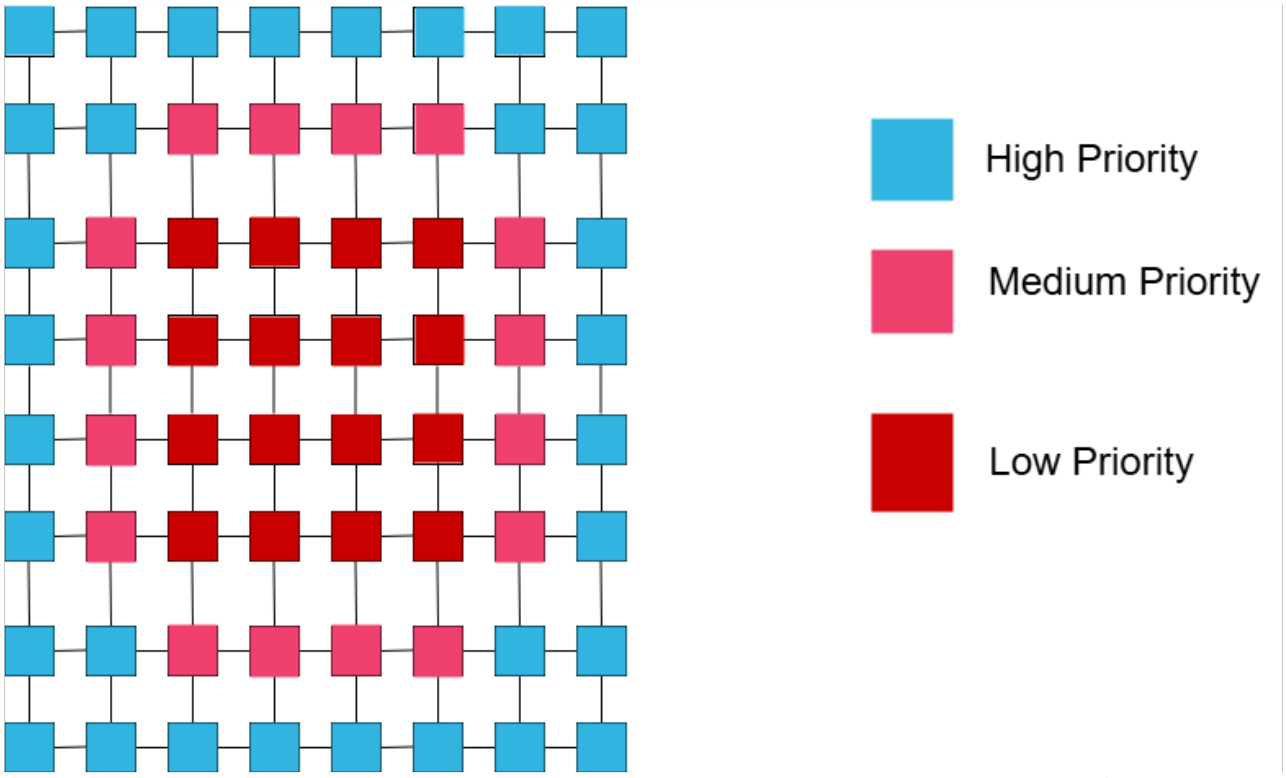



**Figure 7: Marked priority in a $8 \times 8$ mesh according the values of $\tau_1$ and $\tau_2$.**

FreeRider[48], is a non – local adaptive routing strategy was proposed by Liu. S et al. and tested it for mesh NoC topology, which aimed at improving Network-on-Chip (NoC) performance by intelligently regulating congestion, by integrating the congestion information within the head flit of the packet that is propagated during any routine routing exercise. This congestion information is carried by the free bits available in the head flit of a 128 – bit NoC. Thereby eliminating the need for a separate dedicated congestion propagation network (CPN). Congestion information also includes the hotspot areas in the network. A hotspot isdefined as the area or link which experiences a unusually large congestion. A head flit along with the routing information carries hotspot information of k backyard links in the same dimension in which the packet isbeing routed and are called type 1 links. Also, the head flit carries the information of 2k orthogonal links ofall the router it visited during the routing exercise in the same dimension. This hotspot information is saved inthe destination node for future reference and also it gets updated by any other subsequent head flit from thesame direction.



Vol. 23, No. 1, (2026)
ISSN: 1005-0930

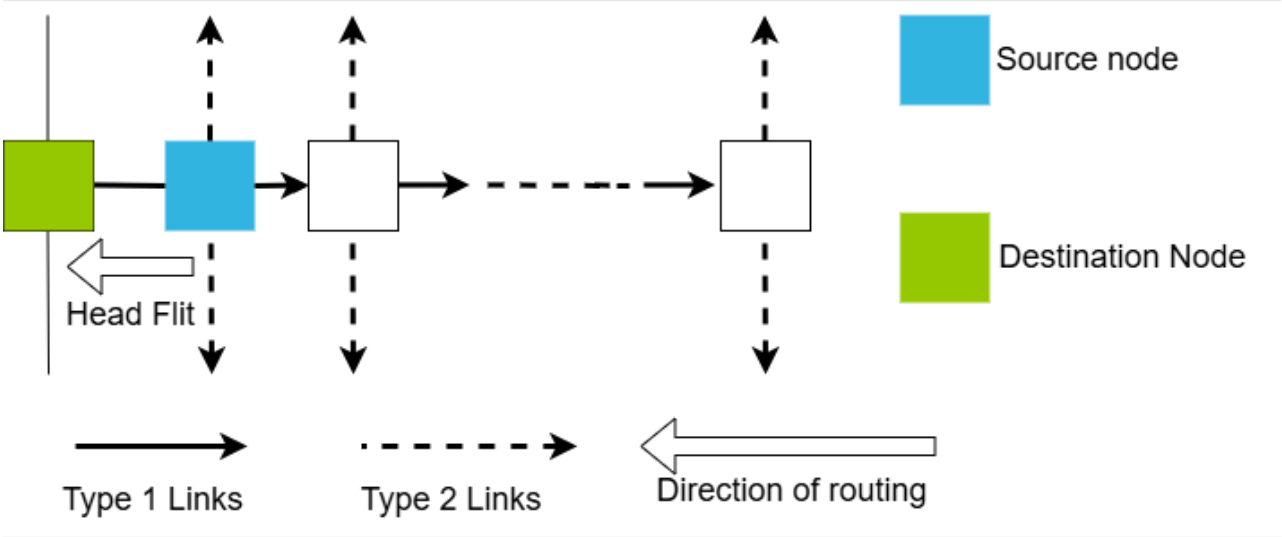



**Figure 8: The routing flow and congestion information flow of FreeRider.**

FreeRider is a three-stage routing algorithm.

**Stage 1**: Initially, the router assesses the local candidate output links to determine if they are exhibiting hotspot/no hotspot characteristics, which are generally signalled by the hotspot/no hotspot bit set to 1/0 for thatparticular link. Links classified as hotspots are then deprioritized, whereas non-hotspot links are favoured tomitigate congestion. Otherwise, if both ports are hotspot/no hotspot i.e. a tie, the routing moves to stage 2 foradecision.

**Stage 2:** If at the first stage an output link is not selected due to tie condition at all the candidate output ports,the second stage examines the k-backyard channels iteratively. If after $i\ th$ iteration the candidate port havingno hotspot condition is found, then the flit is routed along that port, otherwise, if the values are still a tie,wego to the third stage.

**Stage 3:** The stage three is straight forward, when no decision is reached, the routing mechanism looks holistically, the number of vcs occupied in the local candidates ports and the flit is forwarded along the port havi less number of vcs occupied, if at this stage also there is a tie, the flit is routed randomly via any of the candidate ports towards the destination.

Next we summarize the above routing algorithms and based on their nature and discuss the advantages anddisadvantages as reported through various research article.

**Table 1: Summary of Routing algorithms studied**

| Routing Algorithm | Routing policy | Pros | cons |
|---|---|---|---|
| XY | Deterministic | Deadlock free routing[11][49], Minimal-path routing[50], Low implementation | Static routing, poor load balancing[46], Low bandwidth utilization[47] |

| | | | |
|---|---|---|---|
| | | complexity[4], [28] | |
| ROMM[34] | Deterministic | Combines routing flexibility and minimal routing, good average case latency[51], | Poor worst-case latency, Poor load balancing[48] |
| Turn Model[29] | Partially adaptive | Deadlock free routing without complicatingthe router by the use of Virtual channels[52][53]. | Less path diversity, no global congestion knowledge[54]. |
| Odd – Even [32] | Partially adaptive | More path diversitythan other turn models,Even distribution of paths , Deadlock free without virtual channel[54] | Locally adaptive and no global knowledge of congestion.[54] |
| DyAD | Deterministic and adaptive | Freedom from deadlock and livelocks. Flexible path choosing depending upon congestion status[55]. | Throughput declines for certain traffic pattern[56]. No global knowledge of congestion. |
| DyXY | Partially Adaptive | Deadlock free, relatively simple. | Local congestion awareness only, hence choosing a path with congestion ahead is possible[39]. |
| EDXY | Partially Adaptive | Deadlock free and more congestion aware as compared to DyXY. | Extra Hardware is required for handling congestion flag information making routering and router microarchitecture complex. False positive for congestion may be reported during routing[39]. |
| Deflection Routing | Fully Adaptive | Buffer less Router design makes the RTL level router realization easy, cost effective, and consumes less power[41]. | High Latency and low saturation point due to decreased throughput[41]. |
| Context Aware Agent | Deterministic and Adaptive | Deadlock Free[35] | Extra buffer logic is required[35] |
| Neighbor – on – Path [42] | Partially Adaptive | Average Latency is lower than deterministic as it records congestion information from the neighbor which are one hop away in both X and y direction towards the minimal path to destination[57]. | Global information about congestion is not taken into account[48]. Moreover, NoP neglects the congestion in its immediate neighbor, which may result in forwarding the packet towards the congestion which is immediately ahead. |
| RCA (RCA 1D, RCA Fanin, RCA Quadrant) | Fully Adaptive | Congestion awareness from a wider area in the network, relevance of the information is based on its | Routing decisions are still made locally and distributed, which could lead to choices that aren't |



Vol. 23, No. 1, (2026)
ISSN: 1005-0930

| | | | |
|---|---|---|---|
| | | distance from the current node[58]. | the best for the whole system. Moreover, the performance benefit of RCA over basic deterministic adaptive (DA) routing decreases as the system gets larger[58]. Non overlapping information competitates the routing in RCA Fanin[45]. |
| DBAR[45] | Deterministic and Adaptive | Utilizes RCA 1D as base line routing, minimizing non overlapping congestion information by selecting relevant quadrant of Source – Destination pair limited by X and Y axis neighbors[45]. | Requires a separate Congestion propagation Network (CPN) which is bandwidth centric[59]. |
| DAHR[46][60] | Deterministic and Adaptive | No separate CPN is needed, routing information is pre-calculated during injection time[46] | Avoids deadlock using selected turn models restricting adaptivity[60]. |
| ParRouting[47] | Fully Adaptive | Global Knowledge about congestion. Also logs remote node congestion for direction selection based on the priority of the nodes[47]. | Complex priority-based routing doesn't perform well for low network size. |
| FreeRider[48] | Fully Adaptive | Low-cost congestion propagation without CPN. | Complex three stage routing. |

## 4. Conclusion

This study presents the concepts and classification of NoC routing algorithms for mesh architectures. Every kind of routing (oblivious, adaptive, and half-adaptive) has particular characteristics that are tailored to specific real-world applications. This work presents recent mesh NoC research. As additional cores are put on chips, NoCs will play a significant role in interconnection. Routing algorithms are crucial for NoC performance and will continue to be a focus of research.

This study also summarizes the various pros and cons each and every routing strategy studied here with the aim to motivate researchers to address the cons and suggest modifications at the architectural level to meet the modern-day commercial Network – On – Chip requirements such as low power consumption, area minimization, acceptable latency hardware size trade-offs.

**References:**

[1] M. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer* ,vol.35, no. 1, pp. 70–78, 2002.

[2] W. J. Dally, "Virtual-channel flow control,"*IEEETransactionsonParalleland DistributedSystems* , vol. 3, no. 2, pp. 194–205, 1992, doi: 10.1109/71.127260.

[3] W. J. DallyandB.P.Towles,"Routepackets,notwires:on-chipinterconnectionnetworks,"in *Proceedings of the 38th Design Automation Conference (DAC)*, 2001, pp. 684–689.

[4] P. GuerrierandA.Greiner,"Agenericarchitectureforon-chippacket-switchedinterconnections,"in *Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537)*, IEEE Comput. Soc, 2000, pp. 250–256. doi: 10.1109/DATE.2000.840047.

[5] Y. Ajima,S.Sumimoto,andT.Shimizu,"Tofu:A6DMesh/TorusInterconnectforExascale Computers," *Computer (Long Beach. Calif).*, vol. 42, no. 11, pp. 36–40, 2009, doi: 10.1109/MC.2009.370.

[6] K. SrinivasanandK.S.Chatha,"LayoutawaredesignofmeshbasedNoCarchitectures,"in *Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*, New York, NY,USA:ACM,Oct.2006,pp.136–141.doi:10.1145/1176254.1176288.

[7] D. C. Jung, S.Davidson,C.Zhao,D.Richmond,andM.B.Taylor,"RucheNetworks:Wire-maximal, no-fuss NoCs,"in *Proceedings of the 14th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, 2020.doi:10.1109/NOCS50636.2020.9241586.

[8] X. Yu, L. Tang,J.Mi,J.Liu,andL.Long,"Fault-tolerantandqualityofserviceawarerouting algorithm basedonprioritytechniqueforscalablenetwork-on-chiparchitectures," *Sci. Rep.*,vol.15, p. Article 36578,2025,doi:10.1038/s41598-025-20381-3.

[9] L. M. Ni andP.McKinley,"Asurveyofwormholeroutingtechniquesindirectnetworks," *Computer*, *IEEE* vol.26,no.2,pp.62–76,1993,doi:10.1109/2.192002.

[10] J. Duato, "Anecessaryandsufficientconditionfordeadlock-freeroutinginwormholenetworks," *IEEE TransactionsonParallelandDistributedSystems*,vol.6,no.10,pp.1055–1067,1995,doi: 10.1109/71.444850.

[11] W. J. Dally andC.L.Seitz,"Deadlock-freemessageroutinginmultiprocessorinterconnection networks," *IEEE Transactions on Computers*, vol. 36, no. 5, pp. 547–553, 1987, doi: 10.1109/TC.1987.1676939.

[12] C. Wang, W. H.Hu,N.Bagherzadeh,andS.E.Lee,"Areaandpower-efficientinnovativenetwork-on-chip architecture,"in*Proceedingsofthe18thEuromicroConferenceonParallel,Distributedand Network-Based Processing (PDP)*, 2010, pp. 533–539. doi: 10.1109/PDP.2010.15.

[13] A. Kiasari, M.Elsayed,andL.-S.Peh,"Asurveyofexpresslinksinnetworks-on-chip,"*IEEEDes. Test*, vol. 30,no.5,pp.8–21,2013.

[14] A. Kumar, L.-S.Peh,andW.J.Dally,"Expressvirtualchannels:Towardsahigh-performanceNoC architecture," *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1292–1305, 2009, doi: 10.1109/TC.2009.63.

[15] B. Wolfs andK.Goossens,"Expressmeshesforlow-latencyNoCs,"*IEEETransactionsonComputer-Aided Designof Integrated Circuits and Systems* ,vol.33,no.12,pp.1838–1851,2014,doi: 10.1109/TCAD.2014.2369460.

[16] J. Balfour andW.J.Dally,"DesigntradeoffsfortiledCMPNoCarchitectures,"*Proceedingsofthe IEEE InternationalSymposiumonHighPerformanceInterconnects*,vol.14,no.2,pp.69–78,2006.

[17] J. Becker, N.Kumar,andothers,"AhierarchicalNoCforlarge-scaleMPSoCs,"*ACMTransactions on EmbeddedComputingSystems*,vol.12,no.3,pp.1–23,2012.

[18] N. E. Jerger,L.-S.Peh,andM.H.Lipasti,"HierarchicalringNoCsforscalableon-chip communication," *IEEE Micro*, vol. 29, no. 5, pp. 19–31, 2009, doi: 10.1109/MM.2009.84.

[19] K. Kang, L. Benini,andG.DeMicheli,"Cost-EffectiveDesignofMesh-of-TreeInterconnectfor Multicore ClustersWith3-DStackedL2ScratchpadMemory,"*IEEETrans.VeryLargeScaleIntegr. VLSI Syst.*, vol.23,no.9,pp.1828–1841,Sep.2015,doi:10.1109/TVLSI.2014.2346032.

[20] Thompson, "GeneralizedConnectionNetworksforParallelProcessorIntercommunication,"*IEEE Transactions on Computers*, vol. C–27, no. 12, pp. 1119–1125, Dec. 1978, doi: 10.1109/TC.1978.1675014.

[21] R. Miller andQ.F.Stout,"GeometricAlgorithmsforDigitizedPicturesonaMesh-Connected Computer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 2, pp. 216–228, 1985, doi: 10.1109/TPAMI.1985.4767645.

[22] Siegel, "A Model of SIMD Machines and a Comparison of Various Interconnection Networks,"*IEEE Transactions on Computers*, vol. C–28, no. 12, pp. 907–917, 1979, doi: 10.1109/TC.1979.1675280.

[23] W. J. Bouknight, S. A. Denenberg, D. E. McIntyre, J. M. Randall, A. H. Sameh, and D. L. Slotnick, "The Illiac IV system," *Proceedings of the IEEE* , vol. 60, no. 4, pp. 369–388, 1972, doi: 10.1109/PROC.1972.8647.

[24] R. Cypher and J. L. C. Sanz, "Mesh Connected Computers," in *The SIMD Model of Parallel Computation*, New York, NY: Springer New York, 1994, pp. 20–33. doi: 10.1007/978-1-4612-2612-3_4.

[25] A. Hemani, A. Jantsch, A. Postula, J. Berg, and D. Lindqvist, "Network on a Chip: An architecture for billion transistor era," Dec. 2001.

[26] A. Jantsch, L. for Electronics, and R. I. of T. Computer Systems, *Networks on chip*. [Online]. Available: https://jantsch.se/AxelJantsch/papers/2002/rvk-2002.pdf

[27] S. Kumar *et al.*, "A network on chip architecture and design methodology," in *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, IEEE Comput. Soc, pp. 117–124. doi: 10.1109/ISVLSI.2002.1016885.

[28] J. Duato, S. Yalamanchili, and L. M. Ni, *Interconnection Networks: An Engineering Approach*. San Francisco, CA, USA: Morgan Kaufmann, 2003.

[29] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *ACM SIGARCH Computer Architecture News*, vol. 20, no. 2, pp. 278–287, May 1992, doi: 10.1145/146628.140384.

[30] R. R. Bordawekar, A. N. Choudhary, and J. M. del Rosario, "An experimental performance evaluation of Touchstone Delta Concurrent File System," in *Proceedings of the 7th international conference on Supercomputing*, New York, NY, USA: ACM, Aug. 1993, pp. 367–376. doi: 10.1145/165939.166013.

[31] W. Zhang, L. Hou, J. Wang, S. Geng, and W. Wu, "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip," in *2009 WRI Global Congress on Intelligent Systems*, IEEE, 2009, pp. 329–333. doi: 10.1109/GCIS.2009.110.

[32] Ge-Ming Chiu, "The odd-even turn model for adaptive routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 729–738, Jul. 2000, doi: 10.1109/71.877831.

[33] W. J. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 4, pp. 466–475, 1993.

[34] T. Nesson and S. L. Johnsson, "ROMM routing on mesh and torus networks," in *Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures - SPAA '95*, New York, New York, USA: ACM Press, 1995, pp. 275–287. doi: 10.1145/215399.215455.

[35] M. Nickray, M. Dehyadgari, and A. Afzali-kusha, "Adaptive routing using context-aware agents for networks on chips," in *2009 4th International Design and Test Workshop (IDT)*, IEEE, Nov. 2009, pp. 1–6. doi: 10.1109/IDT.2009.5404108.

[36] J. Hu and R. Marculescu, "DyAD: Smart routing for networks-on-chip," in *Proceedings of the 41st Annual Design Automation Conference (DAC)*, 2004, pp. 260–263. doi: 10.1145/996566.996638.

[37] M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY: A proximity congestion-aware deadlock-free dynamic routing method for network-on-chip," in *Proceedings of the 43rd Design Automation Conference (DAC)*, 2006, pp. 849–852. doi: 10.1145/1146909.1147125.

[38] M. Ebrahimi, M. Daneshtalab, J. Plosila, and F. Mehdipour, "MD: Minimal path-based fault-tolerant routing in on-Chip Networks," in *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE, Jan. 2013, pp. 35–40. doi: 10.1109/ASPDAC.2013.6509555.

[39] P. Lotfi-Kamran, A. M. Rahmani, M. Daneshtalab, A. Afzali-Kusha, and Z. Navabi, "EDXY – A low costcongestion-awareroutingalgorithmfornetwork-on-chips," *Journal of Systems Architecture*, vol. 56, no. 7, pp. 256–264, Jul. 2010, doi: 10.1016/j.sysarc.2010.05.002.

[40] Z. Lu, M. Zhong, and A. Jantsch, "Evaluation of on-chip networks using deflection routing," in *Proceedings of the 16th ACM Great Lakes Symposium on VLSI*, in GLSVLSI '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 296–301. doi: 10.1145/1127908.1127977.

[41] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 196–207, 2009, doi: 10.1145/1555815.1555784.

[42] G. Ascia,V.Catania,M.Palesi,andD.Patti,"Anewselectionpolicyforadaptiveroutinginnetwork on chip,"Jan.2006.

[43] P. Gratz,B.Grot,andS.W.Keckler,"Regionalcongestionawarenessforloadbalanceinnetworks-on-chip,"in*2008IEEE14thInternationalSymposiumonHighPerformanceComputerArchitecture*, IEEE, Feb. 2008, pp. 203–214. doi: 10.1109/HPCA.2008.4658640.

[44] F. HassenandL.Mhamdi,"High-capacityclos-networkswitchfordatacenternetworks,"in*2017 IEEEInternationalConferenceonCommunications (ICC)*,IEEE,May2017,pp.1–7.doi: 10.1109/ICC.2017.7997147.

[45] S. Ma,N.EnrightJerger,andZ.Wang,"DBAR:Anefficientroutingalgorithmtosupportmultiple concurrentapplicationsinnetworks-on-chip,"in *Proceedings of the 38th annual symposiumonComputerarchitecture*,NewYork,NY,USA:ACM,Jun.2011,pp.413–424.doi: 10.1145/2000064.2000113.

[46] N. Ji andY.Yang,"ADeadlock-FreeDeterministic–AdaptiveHybridRoutingAlgorithmforEfficient Network-on-ChipCommunication," *Electronics (Basel).* ,vol.14,no.5,p.845,Feb.2025,doi: 10.3390/electronics14050845.

[47] J. Fang,D.Zhang,andX.Li,"ParRouting:AnEfficientAreaPartition-BasedCongestion-Aware RoutingAlgorithmforNoCs," *Micromachines (Basel).*,vol.11,no.12,p.1034,Nov.2020,doi: 10.3390/mi11121034.

[48] S. Lee,*et al.*, "Light-Weight Non-LocalAdaptiveNetwork-on-ChipRoutingwithPacket-Carried PropagationofCongestionInformation,"*IEEETransactionsonParallelandDistributedSystems*,vol. 26, no.8,pp.2272–2285,Aug.2015,doi:10.1109/TPDS.2014.2345065.

[49] J. Duato,"Anewtheoryofdeadlock-freeadaptiveroutinginwormholenetworks,"*IEEETransactions on ParallelandDistributedSystems*,vol.4,no.12,pp.1320–1331,1993,doi:10.1109/71.250114.

[50] L.-S. PehandW.J.Dally,"Adelaymodelandspeculativearchitectureforpipelinedrouters," *Proceedings of the International Symposium on High Performance Computer Architecture*, pp. 255–266, 2001.

[51] Daeho Seo,A.Ali,Won-TaekLim,N.Rafique,andM.Thottethodi,"Near-OptimalWorst-Case ThroughputRoutingforTwo-DimensionalMeshNetworks,"in*32ndInternationalSymposiumon Computer Architecture (ISCA'05)*, IEEE, pp. 432–443. doi: 10.1109/ISCA.2005.37.

[52] R. KAWANO,R.YASUDO,H.MATSUTANI,M.KOIBUCHI,andH.AMANO,"AGeneralized TheoryBasedontheTurnModelforDeadlock-FreeIrregularNetworks," *IEICE Trans. Inf. Syst.*,vol. E103.D, no. 1, pp. 101–110, Jan. 2020, doi: 10.1587/transinf.2018EDP7367.

[53] Z. Zhang,A.Greiner,andS.Taktak,"Areconfigurableroutingalgorithmforafault-tolerant2D-Mesh Network-on-Chip,"in *Proceedings of the 45th annual Design Automation Conference*,NewYork, NY, USA:ACM,Jun.2008,pp.441–446.doi:10.1145/1391469.1391584.

[54] L. Liu, Z.Zhu,D.Zhou,andY.Yang,"AfairarbitrationforNetwork-on-Chiproutingwithodd-even turn model," *Microelectronics J.*,vol.64,pp.1–8,Jun.2017,doi:10.1016/j.mejo.2017.04.002.

[55] M. S. Saliu,M.OmuyaMomoh,P.UchennaChinedu,W.Nwankwo,andA.Daniel,"Comparative PerformanceAnalysisofSelectedRoutingAlgorithmsbyLoadVariationof2-DimensionalMesh TopologyBasedNetwork-On-Chip," *ELEKTRIKA- Journal of Electrical Engineering*,vol.20,no.3, pp. 1–6,Dec.2021,doi:10.11113/elektrika.v20n3.249.

[56] X. Yu,L.Tang,J.Mi,J.Liu,andL.Long,"Faulttolerantandqualityofserviceawarerouting algorithmbasedonpriorityteчniqueforscalablenetworkonchiparchitectures," *Sci. Rep.*,vol.15, no. 1,p.36578,Oct.2025,doi:10.1038/s41598-025-20381-3.

[57] Y. Sun,X.Qian,andX.Qu,"AdaptiveRoutingAlgorithmsofNetwork-on-Chip:ALiterature Review," *Applied and Computational Engineering*,vol.132,no.1,pp.213–225,Feb.2025,doi: 10.54254/2755-2721/2024.20708.

[58] R. Manevich,I.Cidon,A.Kolodny,I.Walter,andS.Wimer,"ACostEffectiveCentralizedAdaptive Routing forNetworks-on-Chip,"in *2011 14th Euromicro Conference on Digital System Design*,IEEE, Aug. 2011,pp.39–46.doi:10.1109/DSD.2011.10.

[59] S. Liu, Y.Chen,T.Chen,L.Li,andC.Lu,"GlobalAdaptiveRoutingAlgorithmWithoutAdditional

Congestion Propagation Network," 2012. [Online]. Available: https://arxiv.org/abs/1208.0384

[60] N. Ji and Y. Yang, "A pre-congestion-awaredeterministic-adaptive hybrid routing (PcaDAHR) algorithm for network-on-chip," *Microelectronics J.*, vol. 164, p. 106811, Oct. 2025, doi: 10.1016/j.mejo.2025.106811.