J Supercomput (2015) 71:4260–4286
DOI 10.1007/s11227-015-1521-x

Master-based routing algorithm and communication-based cluster topology for 2D NoC

Suchi Johari[1] · **Vivek Kumar Sehgal**[1]

Abstract As size of chip is becoming smaller with growth in technology, and due to increase in number of cores, system-on-chip (SoC) becomes very complex. Network-on-chip (NoC) provides best solution to SoC by reducing communication overhead. The basic concern of NoC is the speed, performance and accuracy along with the small size of chip. The existing NoC topologies such as mesh topology, bus topology, torus topology, fat tree topology does not provide optimized performance. In this paper, we have proposed communication-based cluster topology (CBCT), which is based on 2H, i.e., heterogeneous and hybrid, proves to be more efficient topology by providing better performance due to reduction in latency, link utilization and energy consumption involved during communication. In experimental result, CBCT approach is compared with 2D mesh topology and CBCT proves to provide better results in terms of an end-to-end latency, network latency, packet latency, sink bandwidth, loss probability, link utilization and energy consumption of a topology.

Keywords Network-on-chip · Topology · Cluster-based communication graph · Latency · Bandwidth · Energy Consumption

1 Introduction

With developing technology, the developers have modified integrated circuits such that they can accommodate a large number of transistors, leading to improvement of

---

✉ Vivek Kumar Sehgal
vivekseh@ieee.org

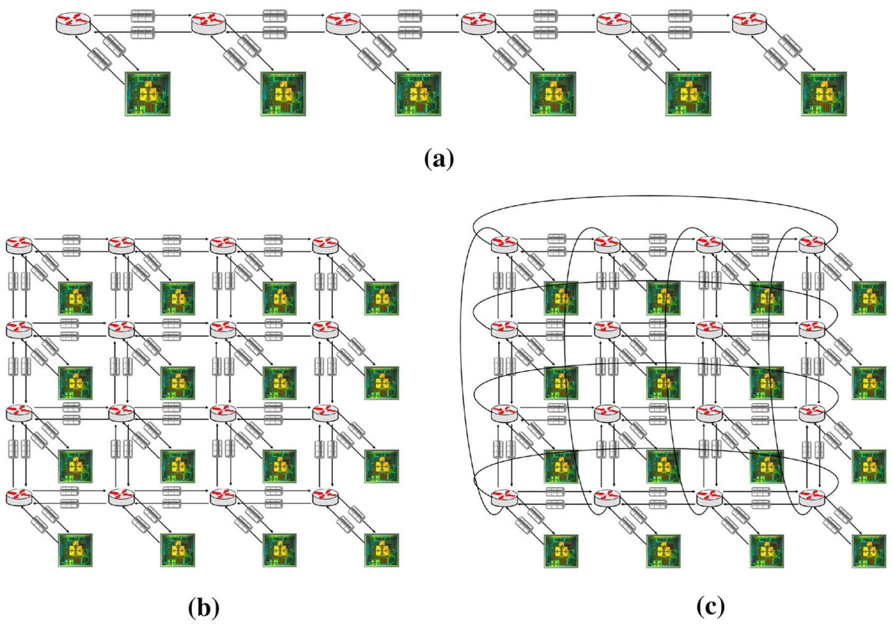
Suchi Johari
shuchi.johari@gmail.com

[1] Department of Computer Science and Engineering, Jaypee University of Information Technology, Solan, India

the computational power of integrated circuits. Size of the circuit is the biggest issue. To increase the number of components for better performance, keeping the size of circuit constant is a big challenge in front of NoC. Traditional methods for connecting the IP cores could not meet the complex structures of the current required NoC. Due to expanding technology, the traditionally accepted interconnection networks are facing problems such as synchronization, latency, energy and power consumption [1–3]. For the better communication among the components in the NoC, the basic requirement is the optimized topology and efficient routing algorithm. Different routing algorithms are proposed by different researchers, which can be classified as source routing, XY routing, west-first routing and OE routing algorithms [4]. NoC provides communication subsystem, which is regular and highly fault tolerant. Appropriate topology and routing algorithm leads to proper communication in the interconnection network which enhances the flexibility and performance of the system [5,6]. The basic concern is the size of NoC, hence during the shrinking of the interconnection network, the latency and power consumption are major factors to be considered [7]. For the purpose of scaling of the integrated circuits, bus topology could not be the sufficient and cannot resolve the issue of scalability of topology [8]. In bus topology, communication is not efficient in terms of scalability, cost and time involved during the communication. Hence the major problem with bus topology is that, as the number of component increases, it adversely affects the latency, link utilization, energy consumption, bandwidth and other parameters. So, the basic concern is to design a topology, which can resolve all these issues. For the scalability of interconnection networks, different approaches are being introduced [9–11] by different researchers. The cost incurred for cores, routers, network interfaces and communication channel determines the cost involved in the development of NoC topology. The communication cost of the NoC topology can be calculated as the number of hops, a message takes during communication. Topology is responsible for the path diversity, i.e., total number of different routes existing between components in the interconnection network [12]. This property of the interconnection network can also be used to determine the bandwidth. In NoC, routing of packet takes place between interconnected components [9]. Different techniques are introduced by different researchers, in which efficient utilization of energy, scalability of the network and high bandwidth are considered [13–15]. But, none of the existing techniques provide guarantee for minimization of the processing time, scalability of the network, minimization of power consumption and energy involved in the optimized bandwidth. If the existing techniques are considered, then to optimize one of the parameter, the other parameters are affected adversely. The main focus of the technique being proposed in this paper, i.e., CBCT is to optimize end-to-end latency, network latency, packet latency, sink bandwidth, loss probability, link utilization and energy consumption, which guarantees faster communication, and high scalability of the NoC topology at minimum bandwidth.

1.1 NoC topologies

NoC can be considered as a circuit consisting of subsystems communicating with each other. Communication typically occurs between routers and IP cores arranged in



(a)

(b)

(c)

Fig. 1 **a** Bus topology. **b** Mesh topology. **c** Torus topology

a particular pattern according to the topology used. NoC can either use synchronous or asynchronous modes of clock for the purpose of communicating between the routers and IP cores. NoC has introduced adaptable network topologies such as mesh, torus and fat tree over traditionally existing bus topology [16,17]. NoC is a layered methodology, where packets are routed between the routers using packet switching. For the purpose of routing of the packets in NoC, two components are required: switches (or routers) and links (or channels). In NoC, topology can be distinguished under two broad categories, i.e., (a) direct topology, where each router or a switch has a core attached to it directly, and (b) indirect topology, in which cores might not be connected directly to some of the routers or switches. Bus topology as shown in Fig. 1a, mesh topology in Fig. 1b and torus topology as in Fig. 1c are examples of the direct topology, in which each router is attached to a core directly, whereas, fat tree topology is an example of indirect topology.

1.2 Routing algorithms in NoC

To communicate between the sender and receiver, routing algorithms are used [18]. Oblivious routing and adaptive routing are two groups for the categorization of the routing algorithms. In oblivious routing algorithms, packets are routed from source to destination without any information, such as the traffic and network information. The oblivious routing algorithm is further subdivided as deterministic and stochastic routing algorithms. Deterministic algorithms are used in the network, where the packets have to be passed following the same route between source and destination, and these routing paths are precomputed [19]. Blocking of traffic in the network is the problem faced in oblivious routing, which can only be resolved by waiting for clearance of traffic, then try to send the packet again. The other name of the adaptive routing is the dynamic routing, where if the condition of the network changes, the route taken for communication between the source and destination also alters dynamically. Common issues, which act as an obstacle in the oblivious and adaptive routing algorithms, are deadlock, livelock and starvation [20,22].

This paper is organized as follows: Sects. 2 and 3 include the related work and problem statement, Sect. 4 provides a detailed description of the proposed approach CBCT along with master-based routing algorithm. In Sect. 5, QoS parameters are discussed. Sections 6 and 7 includes implementation details and experimental results, which proves proposed approach to be better than mesh topology. Section 8 provides conclusion and future work.

2 Related work

As the technology develops, NoC is gaining popularity over the SoCs. Bhandarkar and Arabnia [23] proposed REFINE multiprocessor, a kind of reconfigurable interconnection network that depends upon multi-ring architecture. Bhandarkar and Arabnia [24] performs parallel operations on Hough transform to show reconfigurable multi-ring network (RMRN) [25,26] as a scalable architecture. Arif Wani and Arabnia [27] proposed edge region-based segmentation algorithm performed on reconfigurable multi-ring networks that works parallel.

Chan and Parameswaran [28] has introduced a new approach NoCOUT, which is capable of generating an application which is energy efficient, and can support both types of networks whether point-to-point or packet-switched. For exploring the design space efficiently, the proposed algorithm is iterative in nature which is based on the greedy strategy. The floor planner approach is used to analyze the improvements and feedback for the wire lengths being effected by topology chosen for the network. Li et al. [29] made use of the voltage and frequency for the purpose of achieving efficient energy for the NoC subsystems formed based on the communication. The author has proposed the custom topology which consumes low power, and constructs efficient NoC for irregular topologies. The main aim is to minimization of the energy consumption and improvement of the communication efficiency. Ge et al. [30] has introduced a topology based on clustering approach for the construction of NoC topologies. The major concentration is on the minimization of the power consumption and cost involved. The technique consists of four phases and a custom irregular NoC topology having the constraints related to design, based on the communication requirements and also based on the characteristics of routers. The links between the routers is created by the help of the link construction algorithm which is based upon recursion. Jain et al. [31] has given a topology generator, which is customizable and it has implemented a simulation annealing-based heuristic technique for the purpose of optimization of energy. Srinivasan et al. [32] has introduced a three-phase technique, which is helpful in generating a good performance layout for SoC, also it helps in mapping of the cores to routers, and even it gives a unique path for every task to reach the core. Dafali and

Diguet [33] have given a new technique for generating NoC topologies according to the requirements of the application and the architecture. The topology generated is according to the demands of the designers and the requirements of the application. This approach has customized the technique according to the architecture of multiprocessor and memory organization.

Choudhary et al. [34] has given a technique for the customization of the SoCs irregular networks. The traditional knowledge of application is used, to generate a network in which energy utilization is optimized. Tino and Khan [35] have given a technique based on Tabu Search, in which power consumption is optimized and best performance for providing a better NoC architecture. An automated approach is used to generate the topology, also the details about the floorplan and values like wire length are also calculated. The various issues such as contention, constraints related to performance considered there during topology creation, are discussed. Power and performance metrics are considered for analyzing contention in the network, and for this purpose it makes use of the contention model, i.e., Layered Queuing Network. This contention model is able to analyze all the interaction between components of NoC, and estimate the points in system, where there is the maximum probability of bottleneck occurrence.

3 Problem statement

For reduction of design space, communication delay and cost involved, NoC tries to reduce the number of connections. Reduction in the number of communication channels leads to lack in functionality during traffic and noise. Lesser number of channels between any two nodes lead to the situation of bottleneck, due to which performance is also reduced. So the main concern of this paper is to reduce the end-to-end delay, packet latency, network latency, link utilization and energy consumption, keeping area and bandwidth constraints. The main purpose of this research is to provide a new topology which proves to be better in terms of QoS parameters [36–38].
Given A core communication graph G (or G).
Find The topology generated should optimize O(A,G), subject to the constraints specified by Const(A,G).

- O(A,G) can be a metric for reliability and performance of the network, such as end-to-end latency, network latency, packet network latency, loss probability, link utilization and energy consumption etc.
- Const(A,G) represents the resources (e.g., area, wiring, etc.), cost and bandwidth imposed by the application.

4 Proposed approach

For the implementation of the CBCT, it is required to implement such a routing algorithm which can be adapted according to the topology being used in the cluster. For solving the routing problem, we have proposed a master-based routing algorithm.

4.1 Cluster formation

Cluster in CBCT is formed using the communication-based cluster graph (CBCG). CBCG is an undirected connected graph which can be represented as CBCG $= (C_i, L)$ where, C belongs to C_i represents the clusters where i = 1, 2, 3, 4... and I_l belongs to L represents the inter-cluster links. In CBCG, $w(t_1, t_2)$ is the weight associated with each link which represents the communication between the core t_1 and t_2. In this paper, our main focus is to choose only those cores t_i which have maximum amount of communication, i.e., having more edge weight. So, cores having $\max(w(t_1, t_2))$ are kept in same cluster. In CBCG, the clusters of most communicating cores are formed, the corresponding routers attached to the cores are also considered in the same cluster. Figure 2 shows the communication between the cores. The weight of edges describes the amount of communication taking place between the two cores. For simulation, weights are assigned randomly to links. More the edge weight, more is the communication. According to the amount of communication, clusters can be defined, and communication-based cluster graph is obtained. The clusters are shown in the Fig. 3 as C_1, C_2, C_3, C_4 and the links inside each cluster represent the communication between the cores in the cluster. In Fig. 3, the values associated with the links represent the communication between the cores. These links between the cores of the cluster are called intra-cluster links and the links through which the clusters are connected together are called the inter-cluster links. Cores of two distinct clusters have minimum communication between them. The cores in the CBCG as shown in Fig. 3 can be connected to each other using any topology within a single cluster such that most communicating cores lie within the same cluster. Algorithm 1 gives the details for the formation of clusters. In the Algorithm 1, Let there be 'N' cores in total. Decide the maximum number of clusters required for the topology to be generated as $'T_h'$. Since the core once visited and assigned to some cluster should not be visited again so, they should be kept in an array V[N]. Choose the highest cost $E(x, y)$ among the set of edges and check both the vertices of this edge. E is the set of edges, which represents the communication between the nodes. x represents the first node, whereas y represents the second node, $E(x, y)$ represents the edge between these two nodes to show the communication between the two nodes. If vertex 'x' or 'y' belongs to some cluster, assign the other vertex to the same cluster. If none of the vertex 'x' and 'y' belongs to any cluster then, assign both cores to new cluster. Following the above steps may lead to two major problems (a) number of clusters formed may exceed the threshold decided, (b) clusters formed may be less than the specified number of clusters. If number of clusters formed is less than the specified number of clusters, then choose an edge having the minimum weight. Assign both the cores associated with this edge of a new cluster. The procedure is repeated in a similar manner till the specified number of clusters are not formed. In case more clusters are formed, then take the core from two different clusters which have maximum amount of inter-cluster communication and combine the two clusters to form a single cluster. The complexity of the cluster formation algorithm is $O(n^2)$.

CBCG graph can further be reduced in more generalized graph TBCG. TBCG $=$ $\langle T,t \rangle$, where T represents the topology used to arrange the cores within the cluster and t represents the topology used to connect these clusters. In TBCG as shown in Fig. 4,

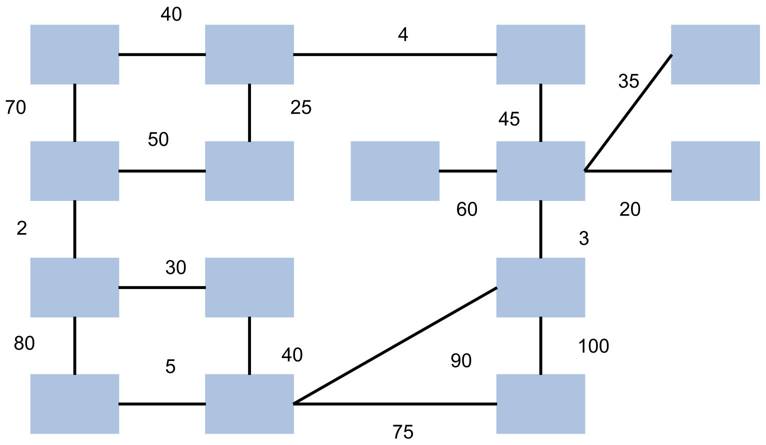


Fig. 2 Communication between cores

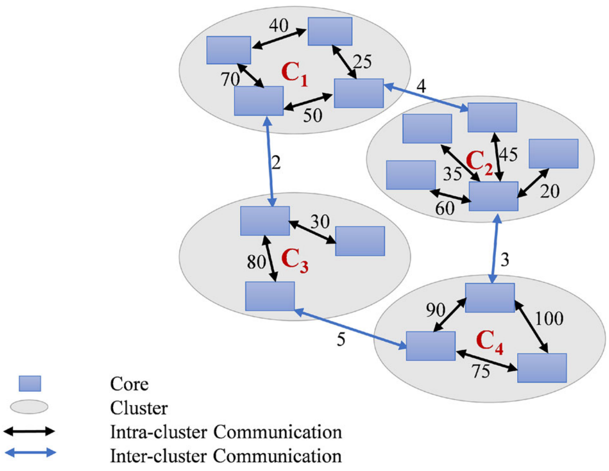


Fig. 3 Intra-cluster and inter-cluster communication in CBCT

T_i represents the topology in cluster C_i and M_i represents the master router for each cluster. Master router is responsible for the inter-cluster communication. From each cluster, a master router is chosen such that it provides connectivity to the outside world, i.e., other clusters. The topologies used within the cluster can vary for each cluster depending upon the requirements. We are dealing with heterogeneous networks, so for proposed approach, we have considered different topology for each cluster. In Fig. 4, the clusters are connected to each other using the mesh topology, but any other topology such as star and bus can also be used to connect the different clusters. Other algorithms such as [30] which form clusters, choose the cores for a cluster randomly and also there are scalability issues as the number of cores per cluster is fixed and also the topology generated by the existing approaches combines the clusters using the bus

Algorithm 1: Cluster Formation for CBCG Graph

```
Data: N = number of cores, {1, ..., n}, $T_h$ = maximum possible clusters, V[N] = visited cores, j=0,
    i=0;
Result: Cluster formation
while $(j! = T_h)$ do
    if i==0 then
        while sizeof(V)!=N do
            Let (x,y) be highest cost edge from E;
            if $((x \in V)\&\&(y \in V))$ then
                E – (x,y);
                Continue;
            else
                if $x \in V$ then
                    Assign core y to the cluster to which x belongs;
                    E – (x,y);
                else
                    if $y \in V$ then
                        Assign core x to the cluster to which y belongs;
                        E – (x,y);
                    else
                        Assign both cores x and y to new cluster $C_i$;
                        E – (x,y);
                        i++;
                    end
                end
            end
            Put cores x and y in V whichever is not present in V;
            Draw an edge between x an y;
        end
    else
        if $j < T_h$ then
            Take edge (x,y) having highest cost where $x, y \in C_i$;
            Assign both cores x and y to new cluster $C_i$;
            i++;
        else
            if $j > T_h$ then
                Take edge (x,y) having highest cost where $\in C_i$ and $\in$ to $C_i$;
                Merge the clusters $C_i$ and $C_j$;
                i–;
            end
        end
        j=i;
    end
end
```

architecture, which has the same number of cores per cluster. Flowchart shown in Fig. 5 describes the steps to be followed for the creation of the CBCT topology.

4.2 Structure of router and core for implementation of CBCT

Router, used to implement proposed approach CBCT, is a 2D router which consists of 5 in-ports and 5 out-ports. In-ports are used by node (router and core) to receive the

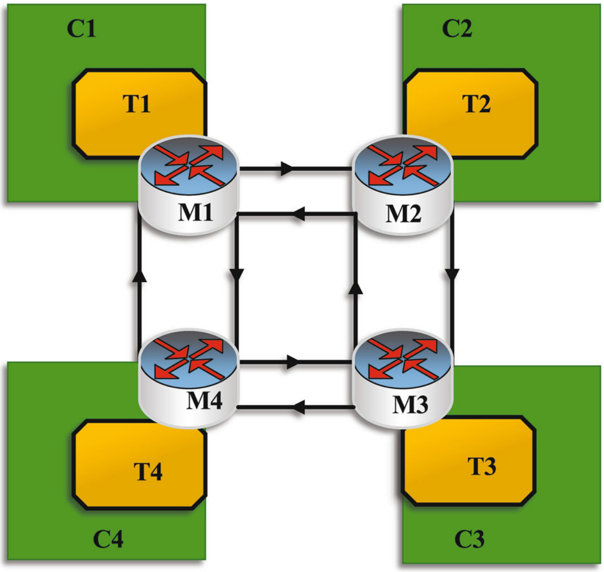


Fig. 4 Topology-based cluster graph

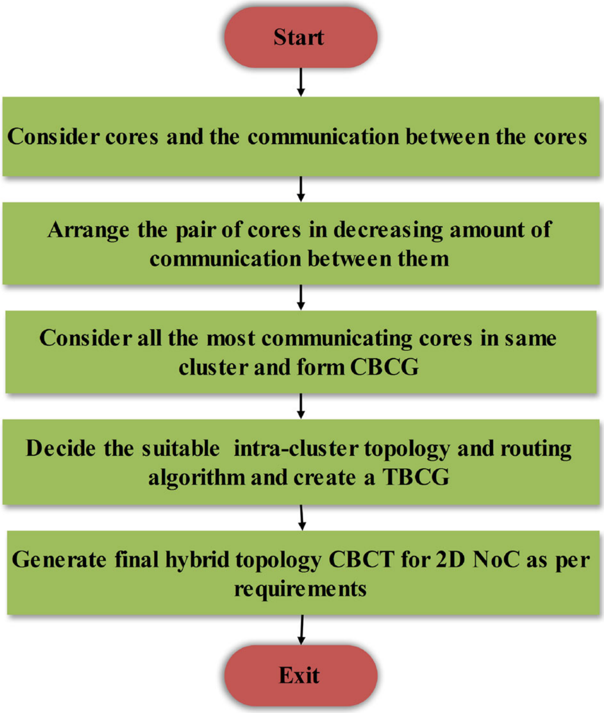


Fig. 5 Flowchart for generating CBCT topology

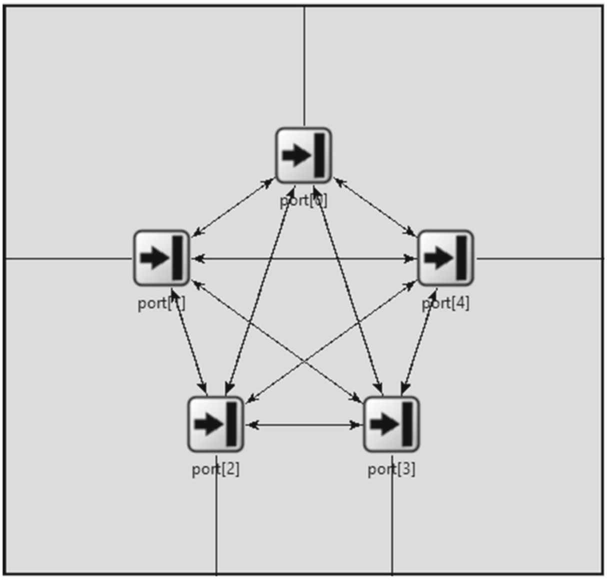


Fig. 6 Structure of 2D router used for CBCT

packet whereas, out-port is used to send the packets to destination. There are 4 ports which are referenced as northPort, southPort, eastPort, westPort to communicate with the routers being connected in the north, south, east and west direction, respectively. Along with these ports, there is one more port, i.e., corePort. This port provides the connectivity among the router and the core. Figure 6 shows the diagrammatic representation of the 2D router being used in the CBCT approach. In Fig. 6, port[0], port[2], port[4], port[1] represent the northPort, southPort, eastPort, westPort, respectively, and port[3] represents corePort. Direction of arrow shows the flow of packets and communication among connected routers. If the structure of the core is considered, then for the implementation of the CBCT, the core is considered as a source as well as the sink as shown in Fig. 7. The two cores involved in the communication act as the source and the sink. Core which transmits the packet to other core acts as the source, whereas the core receiving the packets is referred as the sink.

4.3 Master-based routing algorithm

Existing routing algorithms do not prove to be an optimized solution for the implementation of the proposed approach as the XY routing algorithm is best suited on mesh topology [20] and west-first routing algorithm is suitable for torus topology [21]. But this routing does not work well on the proposed topology, i.e., CBCT so, we have introduced a master-based routing algorithm. In master-based routing algorithm, the routers which are the intermediate between two cluster topologies act as the master routers. For choosing the master node for every cluster, choose the max-

Fig. 7 Structure of 2D core used for CBCT

imum communicating nodes belonging to two different clusters and provide the link between them. Set these two nodes as the master node. Using this approach, a single master for communicating between two distinct clusters is formed. But, it totally depends upon the amount of communication and requirement of the system we can have more than one master in the cluster. As shown in the Fig. 10, yellow-colored routers are the master routers as they are intermediates and they are responsible for the communication between the clusters. These clusters are the networks, which consist of the most communicating components within the same cluster, and lesser communicating in different cluster. As shown in the Fig. 10, consider that there are 3 different topologies used in the 3 different clusters. Cluster C_1 consists of mesh topology, cluster C_2 consists of torus topology and cluster C_3 uses bus topology for connecting different components for communication. For the purpose of routing the packets from source to destination, suitable routing algorithm corresponding to the topology is used like XY routing is best suited to mesh topology, if both source and destination belong to same cluster, whereas if the source and destination belong to different clusters, then master-based routing algorithm comes into active state. In master-based routing algorithm, master has all the information of the routers and cores belonging to same cluster, which is stored in the form of an array during simulation. Routing table can be used for storing the information of inter linked routers. Master also keeps records of the other master router directly or indirectly attached to it. Routers, other than master, also have the record to which cluster that router belongs to and have the information about their master. Suppose source and destination belongs to different clusters, then the packet is passed from the source to the master (corresponding to that cluster to which the source belongs). The master will communicate the other masters for checking to which cluster the destination core belongs to, then it will pass the packet to that master either directly or indirectly. Upon reaching to the master of destination core (following the corresponding routing algorithm associated with the topology of the cluster), the packet will route to the destination core.

Algorithm 2: Master Based Routing

```
Data: Communication Based Cluster Graph (CBCG)
Output: Cluster formation with suitable topology within and between clusters
Initialize $B_{inter}$ and $B_{intra}$;
Set $B_{intra}$ = constant;
$B_{inter} = \frac{(n_t - n_m) * B_{intra}}{n_m}$ ;
while cores are not assigned to some cluster do
    while most communicating cores, belonging to same group in the CBCG, are not all assigned to some cluster do
        Cluster[k] = core[i];
        Choose master core from each cluster and Master[i] = core[j];
    end
end
Assign suitable topology to be used within the clusters and between the clusters according to the requirements;
if both source and destination belong to same cluster then
    Pass packet from source to destination directly;
else
    Pass the packet to the Master core;
end
Master core will further pass it to the other master core and in the same way to other master cores till the cluster having destination core is not reached;
Pass packet successfully to the destination;
```

4.4 CBCT: communication-based cluster topology

To evaluate the performance of CBCT topology, we compared the result of CBCT topology with mesh topology. We have considered 5 × 5 2D mesh topology as shown in Fig. 8, in which the most communicating cores are represented by same colored cubes. Routers are represented as gray-colored circles with a cross.

For simulation of CBCT, we have considered a CBCG as shown in Fig. 9. In Fig. 9, there are four clusters with most communicating cores in the same cluster. These clusters are represented as C_a, C_b, C_c and C_d. Cluster C_a is a combination of most communicating cores $\langle c_0, c_3, c_4, c_6, c_{14}, c_{17}, c_{20}, c_{23}, c_{24}\rangle$, C_b consists of $\langle c_9, c_{10}, c_{11}, c_{12}\rangle$, C_c is a collection of cores $\langle c_2, c_5, c_{22}\rangle$ and C_d consists of $\langle c_1, c_7, c_8, c_{13}, c_{15}, c_{16}, c_{18}, c_{19}, c_{21}\rangle$. If Figs. 8 and 10 are considered, we can conclude that the most communicating cores are far away from each other in a mesh topology, and hence the hop count is large. For example, in Fig. 8, cores 0 and 24 are most communicating cores, and the hop count between these cores is 8, so every time for communication between these two cores, it is required to cover a hop count of 8. This causes an increase in latency with an extra overhead. In CBCT as shown in Fig. 10, the hop count required for the communication between cores 0 and 24 is 4, which is 50 % more efficient than the existing torus topology. Using CBCT approach, generated clustered topology can provide 40–80 % of efficiency over existing topologies. Figure 11 shows a flowchart for creating CBCT topology and routing packets from source to destination.

If two cores 0 and 1 are considered in Fig. 8, it is clear that they are least communicating cores represented using different colors, so keeping these cores far away will

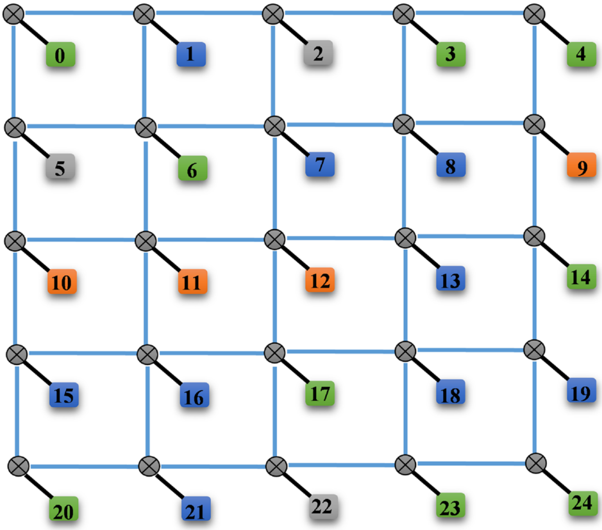


Fig. 8 5 × 5 2D mesh topology

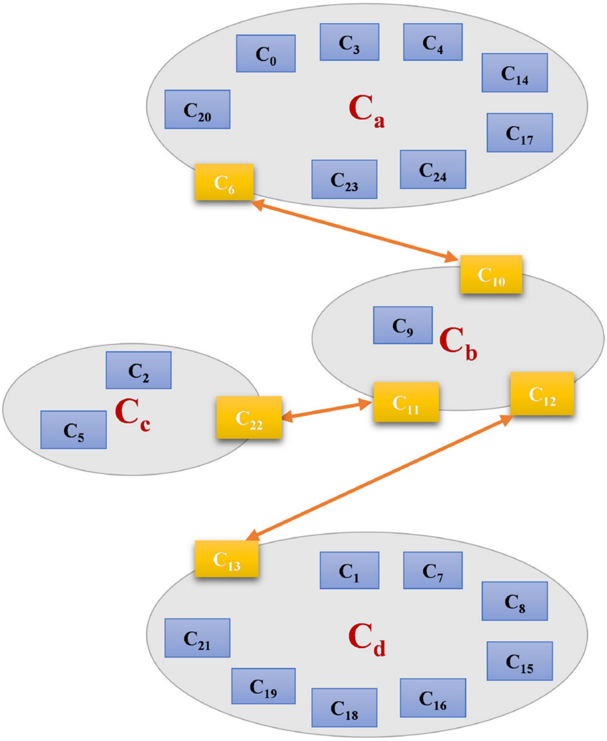


Fig. 9 Cluster of most communicating cores

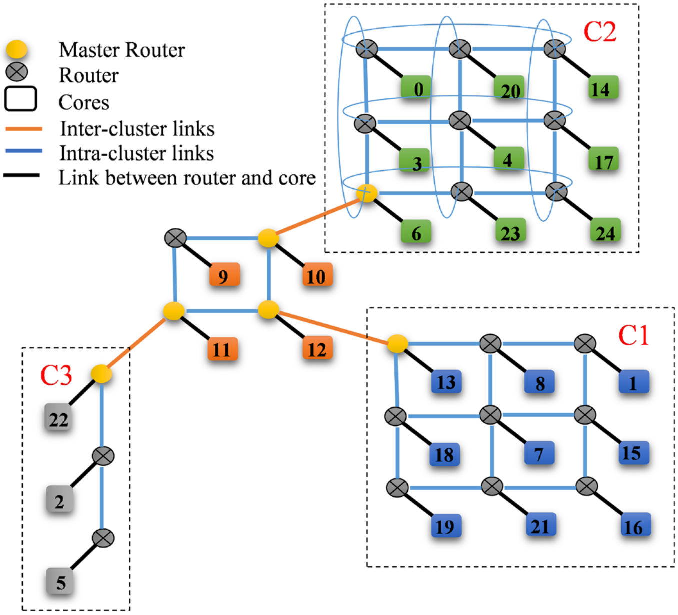


Fig. 10 Communication-based cluster topology

not affect the performance, hence in CBCT as shown in Fig. 10, to bring most communicating cores nearby, these less communicating cores can be placed far from each other. To provide better performance, it is required to optimize latency and overhead. To place all the most communicating cores in one cluster, it should be verified that the hop count between these cores of the cluster should not increase as compared to the hop count between the same cores in a mesh topology. The hop count between the two most communicating cores in the mesh topology is considered as the threshold for this pair of cores in the CBCT. While designing the CBCT, it is required to consider that the threshold determined for a particular pair of cores should be minimized and hop count should be reduced. In the proposed approach, our basic aim is to produce a hybrid topology, which can follow all the QoS requirements such as minimum latency, minimum bandwidth and maximization of the throughput. Communication among the cores can be determined using CBCG graph with the weights assigned to the links. Cores with maximum communication should be in the same cluster, so that they take less time for communication, and hence improves processing speed. Less communicating cores can be in different clusters. Among the nodes within a cluster, one or more nodes are chosen as the master node. Master node act as an intermediate between the two clusters. It is the node responsible for the inter-core communication. The clusters can have any of the NoC topology as per the communication requirements among the cores within the cluster. Since a single cluster can communicate to the multiple clusters, there can be more than one master node within the cluster. If we consider the amount of traffic, then the maximum traffic is found to be at the links joining the two master nodes, as this is the single link which is responsible for the inter-cluster com-

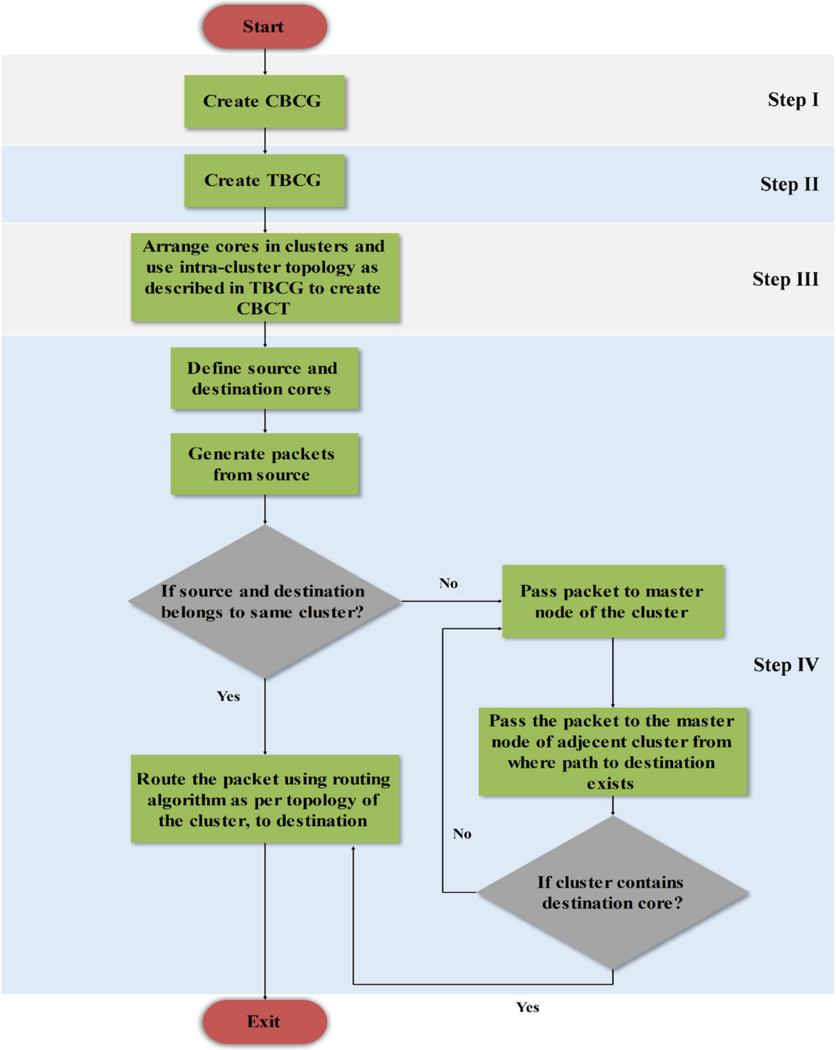


Fig. 11 Flowchart for creating CBCT topology and routing packets from source to destination

munication. To reduce the amount of traffic on these inter-cluster links, it is required that some special configuration of these links has to be specified. Configuration of the links means the delay, bandwidth and the type of wire used as the links. In inter-cluster links, varying the bandwidth of the links has made it possible to simulate the required communication between the cores. B_{intra} be the bandwidth of the intra-cluster links, which is same as the bandwidth of the links of the mesh topology. B_{intra} bandwidth can be adjusted as per the requirements of the architecture, but for the comparison of CBCT with mesh topology during the simulation in this paper we have considered it as 16 GBPS. For this purpose, the mathematical formula used in the proposed approach to set the bandwidth of the inter-cluster links is given as in Eq. 1:

$$B_{\text{inter}} = \frac{N_{\text{nm}} \times B_{\text{intra}}}{N_{\text{m}}}, \tag{1}$$

whereas B_{inter} is the bandwidth of the inter-cluster links, N_{nm} is the number of non-master cores, B_{intra} is the bandwidth of intra-cluster links and N_{m} is number of master cores. This bandwidth reduces the traffic as huge amount of bandwidth is allocated to the most congested links. Depending upon the source and sink, the path is determined in the topology and packets are routed from source to destination using master-based routing algorithm as discussed in Sect. 4.3.

5 Quality of services

For the NoC topology in the proposed approach, we have multiple communication-based clusters. QoS in CBCT topology can be determined by the contribution of the QoS parameters of each cluster individually. We consider the clusters as $\langle C_1, C_2, \ldots, C_n \rangle$, for each cluster a set of QoS parameters will be defined based on the inter-cluster or local topology used in that cluster. Let us consider the QoS parameters as $\langle Q_{i1}, Q_{i2}, \ldots, Q_{in} \rangle$, where i denotes the cluster number. These QoS parameters considered can be normalized to give a threshold value by averaging the minimum and maximum QoS value obtained.

$$Q_{i\_\text{min}} = \min\{Q_{ij}\} \qquad\qquad Q_{i\_\text{max}} = \max\{Q_{ij}\} \tag{2}$$

In Eq. 2, i denotes the cluster number and j denotes the particular parameter for that cluster. We have to consider the mean or the average of two values to get the Q_{th}, i.e., QoS threshold value for the particular cluster as calculated in Eq. 3:

$$\overline{Q_{l\_\text{th}}} = \frac{\sum(Q_{i_{\text{min}}}, Q_{i_{\text{max}}})}{2} \tag{3}$$

QoS value for all the parameters of the cluster is considered and the average is calculated as in Eq. 4:

$$\overline{Q_l} = \frac{\sum_{j=1}^{n} Q_{ij}}{n} \tag{4}$$

This average QoS value obtained decides whether the topology developed is best or not. For the experimental results, we have considered that, if $(\overline{Q_l}) < (\overline{Q_{l\_\text{th}}})$, then the topology generated is not fully optimized, but if $(\overline{Q_l}) > (\overline{Q_{l\_\text{th}}})$, then the topology generated is more optimized. For obtaining a fully optimized global topology, we have introduced the concept of priorities. The parameter which provides Q_{i_max} is considered to be of highest priority and the parameter which provides Q_{i_min} is at the lowest priority. So the QoS values at the local level, i.e., for each cluster, are considered as given in Eq. 5:

$$Q_i = P_{i1} * Q_{i1} + P_{i2} * Q_{i2} + \cdots + P_{in} * Q_{in}, \tag{5}$$

where, P_{ij} determines the priority decided for ith cluster and jth QoS parameter for that cluster. Q_{ij} determines jth QoS parameter for ith cluster. The Q_{ij} obtained for each cluster locally is considered for calculation of the QoS value for full topology globally. The cluster providing the best QoS results is considered to be of highest priority over others. The global QoS value of the whole topology is calculated as in Eq. 6:

$$Q_{\text{total}} = \sum_{i=1}^{n} P_i * Q_i \tag{6}$$

There are certain QoS parameters which should be considered during the generation of the topology for the 2D NoC. We have considered different parameters as follows:

– *Latency* Latency in network-on-chip include the delays involved in the interfaces to pass the packet from router to router, the delays on the routers and the delays due to the traffic, which makes the task to wait inside the core, instead of allowing them to move to the particular core for processing. The formula obtained for the latency in the traditional mesh network would be given by Eq. 7:

$$L_{\text{mesh}} = (n_r * \text{Delay}_r) + (n_l * \text{Delay}_l) + Q_t \tag{7}$$

L_{mesh} represents the total latency in the NoC mesh network. n_r and n_l represent the number of routers and number of transmission links, respectively. Delay_r and Delay_l represent the time taken on the router and time consumed during transmission over the transmission links, respectively. Latency for the proposed topology can be calculated in the similar way. The latency for each cluster is calculated individually at local level, which is summed up to provide the global latency of the whole network. Local Latency of each cluster as in Eq. 8:

$$L_c = (n_{(c,r)} * \text{Delay}_{(c,r)}) + (n_{(c,l)} * \text{Delay}_{(c,l)}) + Q_t, \tag{8}$$

where L_c is the latency of the cluster, c used in the above equation represents the cluster number, r represents router and l represents the transmission links. Global latency of the topology as given by Eq. 9:

$$L_t = \sum_{c=1}^{n} L_c + \sum_{j=1}^{m} \text{Delay}_j \tag{9}$$

Global latency of full network will be the sum of local latencies of all clusters combined with the delay on the inter-cluster communication links. m shows the number of inter-cluster links and Delay_j shows the delay involved on these inter-cluster links.

– *Bandwidth* It is the rate at which data transfer. Network bandwidth should be set such that the maximum data can be transferred either using lesser bandwidth or constant bandwidth. By this a huge amount of data can be passed using comparatively lesser bandwidth, along with minimum latency and energy involved. To get the most optimized results without increasing the amount of bandwidth required, we have to optimize the bandwidth at the inter-cluster links as they are most prone to the traffic and bottleneck. Keeping the bandwidth same at all intra-cluster links as that of the mesh topology, we have increased the inter-cluster link bandwidth proportionally in relation to the intra-cluster links. Let B_{intra} be the bandwidth of the intra-cluster links, which is same as the bandwidth of the links of the mesh topology. B_{intra} bandwidth can be adjusted as per the requirements of the architecture, but for the comparison of CBCT with mesh topology during the simulation in this paper we have considered it as 16 GBPS. B_{inter} is given as in Eq. 10:

$$B_{\mathrm{inter}} = \frac{(n_{\mathrm{t}} - n_{\mathrm{m}}) * B_{\mathrm{intra}}}{n_{\mathrm{m}}}, \tag{10}$$

where n_{t} is the total number of cores, n_{m} be the master cores. So the above relationship is maintained by assigning the bandwidth to the inter-cluster links such that either the total bandwidth required reduces or is constant.

- *Bisection width and bisection bandwidth* The number of links which are broken or removed such that the two networks are obtained, which are equal in size or nearly equal. To obtain multiple paths between these sub-networks obtained, it is required to have a maximum bisection width. More the bisection width, more is the bisection bandwidth. Bisection width in case of $2^n \times 2^n$ 2D mesh is taken as $B_{w\_\mathrm{mesh}} = 2^n$. The bisection bandwidth of the whole network is calculated as given in Eq. 11:

$$B_{b\_\mathrm{mesh}} = B_{w\_\mathrm{mesh}} * \mathrm{Channel}_{\mathrm{bandwidth}} \tag{11}$$

Here, B_{b_mesh} is the bisection bandwidth, B_{w_mesh} is the bisection width of the network and $\mathrm{Channel}_{\mathrm{bandwidth}}$ is the bandwidth of a channel in the network. While calculating the bandwidth in case of the proposed approach, we should divide the network on the inter-cluster links, which will correspond to a network with sub-networks as clusters. So, the bisection bandwidth for the CBCT with 5 interconnected clusters would be 4 and the 5 sub-networks are obtained. So the bisection width of the CBCT topology would be $B_{w\_\mathrm{CBCT}} = l$, where l is the number of inter-cluster links. In CBCT, the bisection width is less, but bisection bandwidth is maximized, hence proves to be better over other existing topologies. As it is considered that to pass a large amount of traffic over the inter-cluster links as they are maximum prone to bottleneck condition, hence to avoid this bottleneck situation, we have considered these links to have more bandwidth over other links. The bisection bandwidth of the whole network would be considered in this case as in Eq. 12:

$$B_{b\_\mathrm{CBCT}} = B_{w\_\mathrm{CBCT}} * \mathrm{ICL}_{\mathrm{bandwidth}}, \tag{12}$$

where B_{b_CBCT} is the bisection bandwidth, B_{w_CBCT} the bisection width of the network and $\mathrm{ICL}_{\mathrm{bandwidth}}$ the bandwidth of inter-cluster link in the network. Suppose that to control the traffic in the network, the bandwidth of the inter-cluster links are taken differently. In that case, the formula in Eq. 12 is given as in Eq. 13:

$$B_{b\_\mathrm{CBCT}} = \sum_{l=1}^{k} \mathrm{ICL}_{\mathrm{bandwidth}}, \tag{13}$$

where k is the number of inter-cluster link and $\mathrm{ICL}_{\mathrm{bandwidth}}$ the bandwidth of the l_{th} inter-cluster link.

- *Loss Probability* Loss probability is the calculation of the lost packet in the stream over the total packets passed. It can be found by dividing the number of packets lost in the network from source to destination by the total number of packets to be sent over the network from source to destination as given in Eq. 14:

$$L_p = \frac{n_t - n_s}{n_t}, \tag{14}$$

where L_{p} is the loss probability, n_{t} the number of total packets to be passed and n_{s} the packets passed successfully over the network. There are multiple paths in the topology generated using proposed approach and also the links having maximum congestion are provided higher bandwidth, to provide less loss probability. During experimental result, we have concluded that the loss probability of the proposed approach is lesser than that of the traditional mesh topology. In mesh topology, the loss probability was calculated as 0.76 whereas in CBCT it is computed to be 0.54.

- *Energy* For defining energy consumption for a NoC topology, combination of cores and routers together is considered as a tile. Energy consumption of packet from tile t_i to t_j is represented as $E_{\mathrm{Packet}}^{t_i,t_j}$. $E_{\mathrm{Packet}}^{t_i,t_j}$ is divided into two parts: (i) E_{Link}, energy consumed by link, and (ii) E_{Router}, energy consumed by the router. E_{Link} is calculated as follows as given in Eqs. 15 and 16:

$$P_{\mathrm{Link}} = (P_{\mathrm{Dynamic}} + P_{\mathrm{Leakage}}) * \mathrm{Num\_Ports} \tag{15}$$

$$E_{\mathrm{Link}} = \frac{P_{\mathrm{Link}}}{\mathrm{Frequency}} \tag{16}$$

E_{Router} is a combination of three energies, i.e., energy of buffer, crossbar and arbitrator as given by Eq. 17. E_{Arbiter} is calculated as given in Eq. 18. $E_{\mathrm{Packet}}^{t_i,t_j}$ from tile t_i to t_j is computed as given in Eq. 19. Total energy consumption of NoC topology (E_{Total}) is calculated in Eq. 20, where N is total number of messages in NoC topologies.

$$E_{\mathrm{Router}} = E_{\mathrm{Buffer}} + E_{\mathrm{Crossbar}} + E_{\mathrm{Arbiter}} \tag{17}$$

$$E_{\mathrm{Arbiter}} = E_{\mathrm{SW\_Allocator}} + E_{\mathrm{VC\_Allocator}} \tag{18}$$

$$E_{\mathrm{Packet}}^{t_i,t_j} = \mathrm{num}_{\mathrm{Hops}} * E_{\mathrm{Link}} + (\mathrm{num}_{\mathrm{Hops}} - 1) * E_{\mathrm{Router}} \tag{19}$$

$$E_{\mathrm{Total}} = \sum_{i=0,j=0}^{N} E_{\mathrm{Packet}}^{t_i,t_j}, \quad \text{where } i \neq j \tag{20}$$

Hence, Eq. 20 is used to evaluate and compare the energy consumption in the case of Mesh topology and CBCT.

6 Implementation details

For the purpose of simulating the CBCT topology for NoC, we have used the OMNET++ simulation tool with HNoC package. We have simulated the hybrid topology (CBCT with 25 nodes) and has compared the results of CBCT topology with 5×5 mesh topology. The parameters considered to get the experimental results for end-to-end latency, network latency, packet latency, sink bandwidth, loss probability and link utilization are given in Table 1 and parameters to calculate energy consumption are given in Table 2. Table 3 gives the detailed description of the values obtained during the simulation of CBCT and Mesh topology in OMNET++. Experimental results in Sect. 6 show that proposed approach proves to be better than other existing topologies.

Table 1 Parameters considered for simulation of mesh topology and CBCT

C_no	Parameters	Mesh topology	CBCT
1	Number of nodes	25	25
2	Rows	5	–
3	Columns	5	–
4	Number of clusters	–	4
5	Routing	$x-y$ routing	According to topology used in cluster
6	Flit size	4 bytes	4 bytes
7	Message length	4	4
8	Packet length	8 (in flits)	8 (in flits)
9	Maximum queued packets	4	4
10	Data rate	4 Gbps	4 Gbps
11	Inter-cluster link	–	3
12	Data rate for inter-cluster link	–	13–16 Gbps

Table 2 Parameters considered for calculation of energy consumption in mesh topology and CBCT

C_no	Parameters	Values
1	Technology used	32 nm(nanometer)
2	Transistor type	LVT
3	Voltage Vdd	1.0 V
4	Frequency	1 × e+9
5	Router in-port	5
6	Router out-port	5
7	Flit width	32 bits
8	Virtual dhannel used	2
9	CrossBar model used	Multistage crossbar switch
10	Buffer size	4
11	Wire type	LOCAL

Table 3 Simulation details for mesh topology and CBCT

C_no	End-to-end latency		Network latency		Packet latency		Sink bandwidth	
	Mesh	CBCT	Mesh	CBCT	Mesh	CBCT	Mesh	CBCT
1	264	152	32	112	88	168	22.725	26.7043
2	280	144	48	104	104	160	20.8313	18.9515
3	296	72	64	32	120	88	20.8313	22.3972
4	312	88	80	48	136	104	30.9313	32.7343
5	264	160	32	120	88	176	21.4625	23.2586
6	280	88	48	48	104	104	20.2	15.5057
7	296	152	64	112	120	168	21.4625	17.2286
8	312	136	80	96	136	152	19.5688	21.5357
9	328	112	96	72	152	128	20.8313	23.2586
10	280	96	48	56	104	112	22.725	24.12
11	296	120	64	80	120	136	19.5688	18.09
12	312	112	80	72	136	128	20.8313	22.3972
13	327.947	120	96	80	152	136	18.0301	14.6443
14	344	72	112	32	168	88	20.8313	20.6743
15	296	168	64	128	120	184	18.9375	17.2286
16	312	184	80	144	136	200	16.4125	18.09
17	328	88	96	48	152	104	17.0438	18.09
18	344	136	112	96	168	152	18.3063	18.9515
19	360	152	128	112	184	168	25.8813	21.5357
20	312.039	72	80	32	136	88	24.2637	25.1968
21	328	168	96	128	152	184	22.0938	20.6743
22	344	128	112	88	168	144	20.2	22.3972
23	360	104	128	64	184	120	18.9375	18.09
24	376	104	144	64	200	120	17.0438	18.09

7 Experimental results

The main purpose of the proposed approach is to optimize the parameters such as end-to-end latency, network latency, packet latency, loss probability, link utilization and energy consumption of topology at minimum bandwidth required. For experimental results, we have considered 25 cores, which can be increased to any number of cores as per the requirements. In the Fig. 12, end-to-end latency of the mesh topology and end-to-end latency of CBCT are compared for different cores keeping source core constant as core 0. Figure 12 clearly shows that CBCT approach is better than existing mesh topology and optimizes approximately 50 % of end-to-end latency.

If we consider the network latency, then the proposed approach proves to be more efficient. To get a clear picture, we can see the graph of network latency for mesh topology and CBCT is compared in Fig. 13. If the average of the network latency and packet latency in both the cases is considered, then it is concluded that CBCT

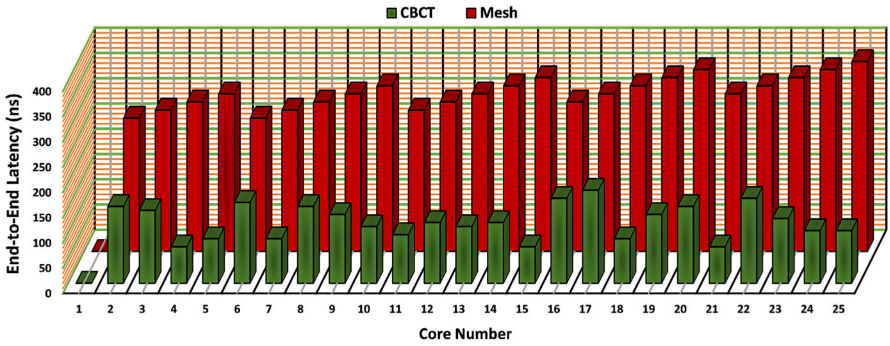


Fig. 12 End-to-end latency

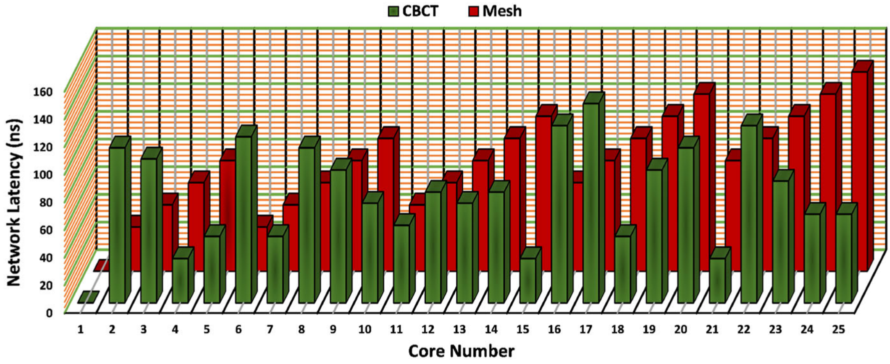


Fig. 13 Network latency

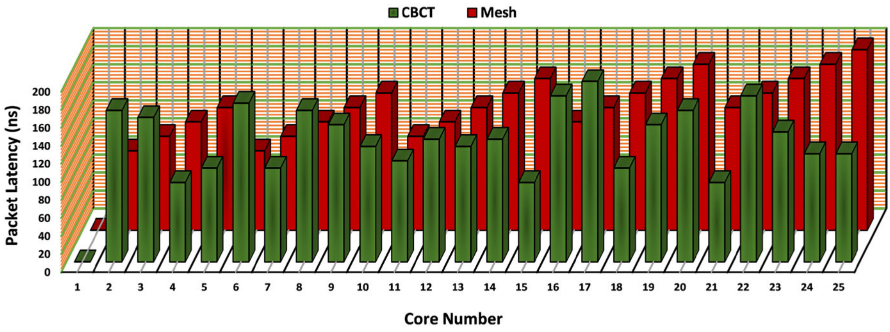


Fig. 14 Packet latency

proves to be more optimized as compared to the mesh topology. These comparisons can be easily visible in the Fig. 14. While to optimize QoS parameters, it is required that bandwidth should not be affected adversely, either bandwidth is optimized or it should remain constant. While simulating the results, keeping the bandwidth constant, we conclude that the results are optimized in CBCT. On the other hand, optimization of bandwidth to a certain extent can produce even better results. Figure 15 shows the bandwidth required in case of mesh topology and CBCT.

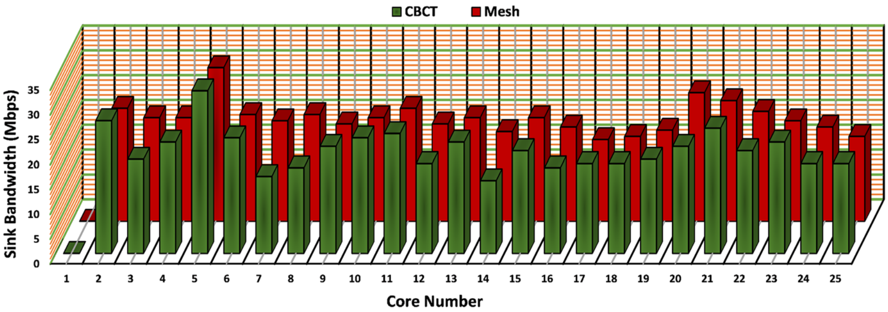


Fig. 15 Sink bandwidth

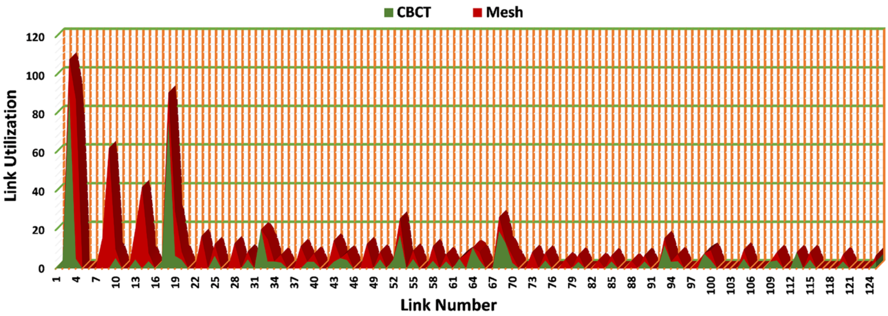


Fig. 16 Link utilization

Link utilization can be defined as the percentage of the total usage of the bandwidth of a particular link. More the link utilization, more is the requirement of bandwidth, leading to a high consumption of energy. Figure 16 shows the comparison of the link utilization for 125 links for 25 cores. From Fig. 16, it is clear that the link utilization in case of CBCT is much lesser as compared to that of mesh topology and hence the least amount of bandwidth is required. Readings for different experimental results are taken in OMNET++ and it was observed that hybrid topology CBCT proves to be better. Its performance can further be improved according to the requirements by customizing the topology for the clusters accordingly.

For calculating the energy consumption in topologies as mentioned in this paper, we have used Orion 2.0 simulator. E_{Link} and E_{Router} at different load and link length (in mm) are calculated in Orion 2.0 simulator and the detailed overview is given in Tables 4 and 5. Using the formula as given in Eqs. 16–20, we compute the energy consumption of mesh and CBCT topology. In Fig. 17, we compared the energy consumption at different cores in both mesh topology and CBCT. Experimental results shows that there is less energy consumption in CBCT topology as compared to mesh topology. CBCT has high speed of data transfer at less bandwidth required in case of most communicating cores. For simplicity, to represent the energy consumption of the topologies, we have considered six cases. Six cores are considered randomly to evaluate energy consumption on each one of them. The cores selected randomly are c_4, c_9, $c_1 4$, $c_1 7$, $c_1 9$, $c_2 4$. Energy consumption for these cores at different link length and load is shown in the Fig. 17. From Fig. 17, it is clearly visible that energy con-

Table 4 Energy consumption of link (in pJ)

Load	Link length					
	1 mm	2 mm	3 mm	4 mm	5 mm	6 mm
0.2	7.65	15.31	22.97	30.63	38.28	45.94
0.4	12.10	24.20	36.30	48.4	60.50	72.6
0.6	16.54	33.08	49.62	66.17	82.71	99.25
0.8	20.98	41.97	62.95	83.94	104.93	125.91
1	25.42	50.085	76.28	101.71	127.14	152.57

Table 5 Energy consumption of router (in pJ)

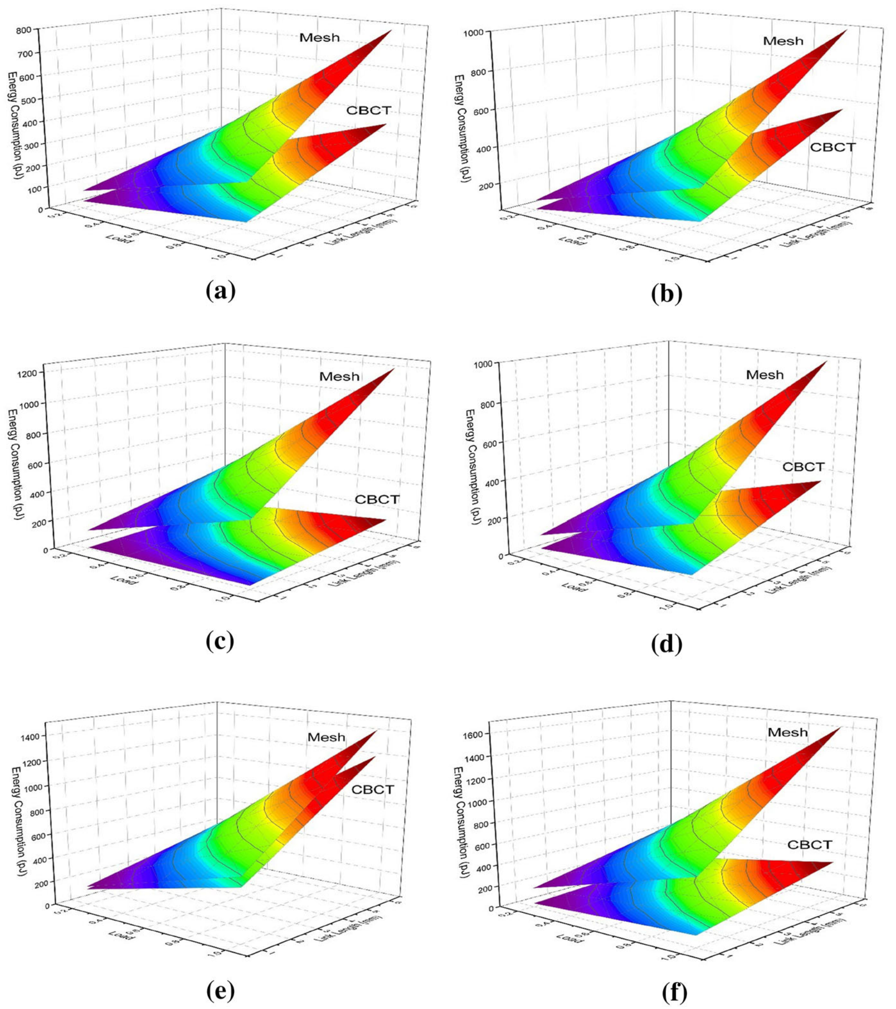
Load	Energy consumption
0.2	16.8
0.4	27.1381
0.6	37.4573
0.8	47.7765
1	58.0958

sumption in CBCT is much lesser as compared to mesh topology for different load and link lengths.

For evaluating the robustness of CBCT over mesh topology, consider a situation in which two cores communicate a large number of times. In case of CBCT, the most communicating cores will be the part of the same cluster making the communication among these two cores faster. But in mesh topology, there is no guarantee of these two cores being near to each other and due to large amount of interaction making the communication paths congested and leading to situation of bottle neck. Since there are very less number of hops involved in the CBCT for most communicating cores, processing speeds up leading to less congestion.

8 Conclusion and future work

CBCT approach proves to be better than the mesh topology in terms of end-to-end latency, network latency, packet latency, loss probability, link utilization, energy consumption of topology and processing speed at minimum bandwidth required. Using CBCT, the hybrid topology generated proves to be 40–80 % more efficient than the existing NoC topologies. CBCT provides the researchers an opportunity to customize the topology according to the requirements of the system. Based on the communication between the cores, most communicating cores are kept in the same cluster. Depending on the number of cores and communication between the cores of the cluster, topology for the cluster is decided. According to the chosen topology, best suited routing algorithm is used for that cluster. If different permutations and combinations of the cores are used to build a cluster, then different level of optimized topology can be obtained. So, this approach provides an opportunity to develop most optimized hybrid and heterogeneous topology as per the conditions and requirements.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Fig. 17 Energy consumption on cores **a** c_4, **b** c_9, **c** c_{14}, **d** c_{17}, **e** c_{19}, **f** c_{24}

As future work, we plan to compare torus, hypercube and fat tree topology with CBCT approach for 2D NoC to prove that CBCT is best approach. Our next focus would be to develop a clustered 3D hybrid and heterogeneous topology which can be customized to produce most optimized results as per the requirements and compare it with 3D mesh, torus, hypercube and fat tree topology.

References

1. Agarwal M, Dubey R, Jain N, Raghuvanshi D (2013) Comparative analysis of different topologies based on network-on-chip architectures. Int J Electron Commun Eng 6:29–40
2. Owens JD, Dally WJ, Ho R, Jayasimha DNJ, Keckler SW, Peh L-S (2007) Research challenges for on-chip interconnection networks. IEEE Micro 5:96–108

3. Sehgal V (2014) Markovian models based stochastic communication in networks-in-package. IEEE Trans Parallel Distrib Syst 26(10):2806–2821
4. Ingle VV, Gaikwad MA (2013) Mesh topology of NoC architecture using source routing algorithm. Int J Eng Adv Technol (IJEAT) 2:101–104
5. Benini L, De Micheli G (2002) Networks on chip: a new SoC paradigm. IEEE Comput 31(1):7078
6. Pavlidis VF, Friedman EG (2007) 3-D topologies for networks-on-chip. IEEE transactions on very large scale integration (VLSI) systems 15:1081–1090
7. Choudhary N (2012) Network-on-chip: a new SoC communication infrastructure paradigm. Int J Soft Comput Eng (IJSCE) 1:332–335
8. Sehgal VK, Chauhan DS (2010) State observer controller design for packets flow control in networks-on-chip. J Supercomput 54(3):298–329
9. Benini L, De Micheli G (2002) Energy-efficient system-level design, Springer book power aware design methodologies, chapter 16. Springer, New York
10. Bertozzi D, Benini L, De Micheli G (2004) Energy-efficient network-on-chip design. In: Macii E (ed) Ultra low-power electronics and design. Kluwer, Dordrecht, pp 214–232
11. Marculescu R, Ogras UY, Peh L-S, Jerger NE, Hoskote Y (2009) Outstanding research problems in NoC design: circuit-, microarchitecture-, and system-level perspectives. IEEE Trans Comput Aided Des Integr Circuits Syst 28:3–21
12. Chen J, Li C, Gillard P (2011) Network-on-Chip (NoC) topologies and performance: a review. In: Proceedings of the 2011 Newfoundland Electrical and Computer Engineering Conference (NECEC), pp 1–6
13. Baboli M, Husin NS, Marsono MN (2014) A comprehensive evaluation of direct and indirect network-on-chip topologies. In: Proceedings of the 2014 international conference on industrial engineering and operations management, pp 2081–2090
14. Patel S, Parandkar P, Katiyal S, Agrawal A (2011) Exploring alternative topologies for network-on-chip architectures. Int J Inf Technol 3:372–376
15. Abou El Hassan BENYAMINA, Pierre BOULET (2007) Multi-objective mapping for NoC architectures. J Digit Inf Manag 5:378–384
16. Choudhary N (2013) Migration of on-chip networks from 2 dimensional plane to 3 dimensional plane. Int J Eng Adv Technol 2:516–519
17. Duato J, Yalamanchili S, Ni L (2003) Interconnection networks: an engineering approach. Elsevier, Amsterdam 2003
18. Xu Y, Zhou J, Liu S (2011) Research and analysis of routing algorithms for NoC, 3rd international conference on computer research and development (ICCRD), pp 98–102
19. Ingle VV, Gaikwad MA (2013) Review of mesh topology of NoC architecture using source routing algorithms. Int J Comput Appl 4:30–34
20. Chawade SD, Gaikwad MA, Patrikar RM (2006) Design XY routing algorithm for network-on-chip architecture. Int J Res Dev 1(1):1–5
21. Rohini A (2013) Design and implementation of minimal adaptive west first algorithm for NoC router architecture. In: Proceedings of the 2013 National Conference on 'Women in Science & Engineering' (NCWSE), pp 269–273
22. Hansson A, Goossens K, Radulescu A (2007) Avoiding message-dependent deadlock in network-based systems on chip, VLSI design. Hindawi publishing corporation, Cairo
23. Bhandarkar SM, Arabnia HR (1995) The REFINE multiprocessor—theoretical properties and algorithms. Parallel Comput 21(11):1783–1805
24. Bhandarkar SM, Arabnia HR (1995) The hough transform on a reconfigurable multi-ring network. J Parallel Distrib Comput 24(1):107–114
25. Bhandarkar SM, Arabnia HR, Smith JW (1995) A reconfigurable architecture for image processing and computer vision. Int J Pattern Recognit Artif Intell (IJPRAI) 9(2):201–229 (special issue on VLSI Algorithms and Architectures for Computer Vision, Image Processing, Pattern Recognition And AI)
26. Arabnia HR, Bhandarkar SM (1996) Parallel stereocorrelation on a reconfigurable multi-ring network. J Supercomput 10(3):243–270
27. Arif Wani M, Arabnia HR (2003) Parallel edge-region-based segmentation algorithm targeted at reconfigurable multi-ring network. J Supercomput 25(1):43–63
28. Chan J, Parameswaran S (2008) NoCOUT : NoC topology generation with mixed packet-switched and point-to-point networks. In: Proceedings of the 2008 Asia and South Pacific design automation conference, pp 265–270

29. Li C-L, Lee JH, Yang J-S, Han TH (2014) Communication-aware custom topology generation for VFI network-on-chip. IEICE Electron Express 11:1–8
30. Ge F, Wu N, Qin X, Zhang Y (2011) Clustering-based topology generation approach for application-specific network on chip. In: Proceedings of the world congress on engineering and computer science, vol 2 (2011)
31. Jain S, Choudhary N, Singh D (2014) STG-NoC: a tool for generating energy optimized custom built NoC topology. Int J Comput Appl 15:22–26
32. Srinivasan K, Chatha KS, Konjevod G (2005) An automated technique for topology and route generation of application specific on-chip interconnection networks. In: IEEE/ACM international conference on computer-aided design, pp 231–237
33. Dafali R, Diguet J-P (2010) MPSoC architecture-aware automatic NoC topology design. Netw Parallel Comput 6289:470–480
34. Choudhary N, Gaur MS, Laxmi V (2011) Energy efficient network generation for application specific NoC. Glob J Comput Sci Technol 11(16):47–56
35. Tino A, Khan GN (2011) Multi-Objective tabu search based topology generation technique for application-specific network-on-chip architectures. In: Design, automation & test in Europe conference & exhibition (DATE), pp 1–6
36. Vellanki P, Banerjee N, Chatha KS (2004) Quality-of-service and error control techniques for network-on-chip architectures. In: Proceedings of the 14th ACM great lakes symposium on VLSI, pp 45–50
37. Zhanga N, Huaxi G, Yang Y, Fan D (2014) QBNoC: QoS-aware bufferless NoC architecture. Microelectron J 45(6):751–758
38. Li B, Zhaoa L, Iyer R, Peh L-S, Leddige M, Espig M, Lee SE, Newell D (2010) CoQoS: coordinating QoS-aware shared resources in NoC-based SoCs. J Parallel Distrib Comput 71(5):700–713