Heterogeneous and Hybrid Clustered Topology for Networks-on-Chip

Suchi Johari, Arvind Kumar, Vivek Kumar Sehgal
Department of CSE and IT
Jaypee University of Information Technology, Solan, H.P., India
{shuchi.johari, er.arvindkumar1989}@gmail.com, vivekseh@ieee.org

Abstract—Reduction in the latency (end-to-end latency and network latency), loss probability, energy consumption and response time are the basic parameters, which are considered by the researchers for the optimization of the networks-on-chip topologies. Different cores communicate at different rates with each other. Some cores communicate at a very rapid rate whereas some rarely communicate to each other. Considering the communication between the cores as the basic parameter, the aim of the paper is to modify the mesh topology and generate a heterogeneous and hybrid clustered topology (HHCT) based on communication between the cores. In the proposed approach, the optimization of latency and response time are major concern, while keeping the geometry of the proposed topology same as that of mesh topology, such that the area required by the topology is constant.

Keywords—Networks-on-chip; heterogeneous topology; Core communication graph; clustered topology.

I. INTRODUCTION

Topologies plays an important role in the networks-on-chip architecture and design. In this section, different existing NoC topologies [1] would be discussed along with their advantages and drawbacks. Ring topology basically considers that each node just has two neighbors and the nodes are connected in such a way that they form an ring structure. In this paper, the term node refers to the combination of core and router. The benefits of ring topology is that, the faults in the network can be easily detected and even it is very cheap and an easy task to configure the ring topology [2]. Ring topology cannot provide a good solution, as in this a single failure of the component may lead to whole system failure, and also the ring topology doesn’t provide scalability. In the star topology, all the nodes are point to point connected to one of the node [3] [4] [5]. If there are ‘n’ number of nodes in the topology, then the degree of one of the nodes acting in the center of the topology is (n-1) and all the other nodes have degree 1. The diameter is constant and is always 2, hence the number of hops are less and due to this latency involved is also less. If any of the node other than the center node fails, then it will not affect the whole system. But the biggest drawback of this topology lies in the central node. Central node has the most congestion and if the central node fails, the whole system will fail altogether. Bus topology used for the networks-on-chip architecture connects the nodes of the NoC in the series. With each node connected to the other in the form of a bus. It is easy to implement, but scalability is difficult to be achieved in the bus topology. Also failure of the single node is difficult to manage.

Mesh topology is the most commonly used topology in the networks-on-chip architecture by the researchers. In the mesh topology, the nodes are arranged in such a way that they form a grid [6]. Mesh topology provides the scalability at a very large amount, but this leads to a disadvantage that the diameter of the topology may be very large. There can be different distinct paths between the nodes in the mesh topology and hence provides a high path diversity. Torus topology can be considered as the extension to the mesh topology. The end nodes in the single row or in the single column are connected to each other directly [7] [8].

All the topologies discussed above are the direct topologies with a direct connection between the nodes in the topology. There exist one more category of topologies which is called as indirect topology. There are a large number of the indirect topologies, such as fat tree topology. In the fat tree, instead of connecting nodes directly to the other nodes, they are connected together with the help of the series of routers. As a tree like structure is formed in the fat tree topology and hence it is called fat tree topology [9] [10].

A. Performance Metrics

There are certain parameters which are chosen by us to determine the performance of the overall topology during this research. These parameters include:

1) End-to-end Latency: It is the amount of time taken from the packet to pass from source to destination i.e. from one end point to another, such that one way communication takes place.

2) Network Latency: It is the latency calculated either in one direction or in both the direction in a network. In one-way, the latency involved from source to destination is considered. In the two-way network latency, the latency involved is calculated as the sum of the time taken for passing the packet from source to destination and the time involved for passing it back to the source.

3) Loss probability: The amount of traffic passed from the source and the amount of traffic received at the destination should be equal. In case, the traffic received at the destination is lesser than the amount of traffic delivered, that means there is some loss of packet. So loss probability is defined as the amount of packets lost to the amount of transferred packets [11].

4) Response time: It is the time taken by the processor before processing any task in the queue. So it is the time, processor required to respond to any request.

5) Energy Consumption: It is defined as the amount of energy consumed by the network during the passing of the packets from one node to the other in network.
II. RELATED WORK

The topologies chosen for networks-on-chip architecture effects the power consumption of the network at a great amount [12]. If the proper routing algorithm is chosen for the networks-on-chip architecture, then it can provide deadlock free transfer of packets leading to the better performance of the network and hence power consumption will also be reduced [13]. To provide an efficient solution to the communication problem being faced in the on-chip networks, researchers have introduced networks-on-chip architecture. Author in the paper [14] has proposed a new topology which is commonly known as Tmesh, in which the standard mesh network is modified by inserting 4 dedicated long links between the nodes which are far away from each other. In the paper [15], the author has proposed an extension over the mesh topology to provide better communication, and has named the proposed approach as the mesh connected crossbar. Author has compared his proposed approach with traditional mesh and torus topologies and has concluded that it produces better results over the other two existing topologies.

Packet-switched NoCs used in multi-core systems used for efficient communication [16]. For providing better communication and to reduce the latency involved, the author in the paper [17] has used different combination of wire configurations for interconnecting the nodes of the topology, such that the communication between the inter-connected nodes improves to a great extent. Authors in the paper [18] have proposed the hybrid-communication reconfigurable NoC, which is proposed in such a way that on the basis of the traffic in the bus topology the MPSoC can configure themselves such that the best solution can be provided. TDMA approach is used by the author to evaluate the performance of the network in the communication. Approach discussed in paper [19] has made it possible to generate the topology based on the communication between the cores such that the energy consumption of the whole network is reduced. In the paper, author has also discussed about the usage of the genetic algorithm to produce the most efficient topology for networks-on-chip. Authors of the paper [20] has proposed the ring based topology for networks-on-chip, which is capable to provide the best results in terms of the latency involved during the communication between the nodes. In paper, the author has discussed about the tracking of the packets in the routing elements and switches. So it would be easy to analyze the tracking of the packets in the network. In paper, author has focused on the introduction of the networks-in-package where the main concentration is on the power and the performance of the system.

This paper is basically divided into certain sections which include the section III as the problem statement, then the proposed approach along with the algorithm is discussed in section IV. Section V gives the detailed explanation of the experimental setup along with the results obtained during simulation of the proposed approach. Section VI gives the detailed description about experimental result and conclusion along with the details of the future work is given in section VII.

III. PROBLEM FORMULATION

Given a core communication graph $G (V, E)$, where $V$ is the set of the vertices and $E$ is the set of the edges, which represents the logical communication between the cores and a mesh topology. The edge weights shows the amount of communication between the particular pair of cores.

Topology generated should be optimized based on $O (A, G)$ keeping some of the parameters constant $C (A, G)$.

Where, $O(A,G)$ determine the parameters to be optimized such as latency, loss probability, link utilization, sink bandwidth and energy consumption, whereas $C(A, G)$ are the parameters which should remain constant. The constant parameter being considered is the geometry is the geometry of the topology. It implies that during the research the mesh topology should be modified in such a way that the equivalent heterogeneous clustered topology formed should not disturb the geometry of the original mesh topology.

IV. PROPOSED APPROACH

The main focus is to make clusters of the most communicating cores together, and then assigning the topology to each cluster based on the communication requirements between them. Our main emphasis is to customize the mesh topology without effecting the geometry and diameter of the mesh topology. Clusters can overlap each other, in case if there is any core which is most communicating for the two other cores belonging to two different clusters. As it is most communicating, hence it will be the part of both the clusters. More the overlapping, more are the chances of congestion. So while designing the clustered topology on the basis of communication, one should always consider that least amount of overlapping among the clusters should be there. The routers, which are responsible for the inter cluster communication are called collaborators. Collaborators are so called because of their functioning, they help the cores from one cluster to communicate with the cores in the other cluster. Collaborators are most critical members of this customized clustered topology. Some benefits are provided to the collaborators in the proposed approach. Since the most of the traffic congestion is found on the links between the collaborators, hence in the proposed approach these links called as inter clustered links are given with higher bandwidth as compared to other links in the topology. While designing the customized topology by the proposed algorithm, designer should keep in mind that increasing or decreasing the bandwidth of the inter and intra cluster links should not affect the overall bandwidth of the whole network adversely. More than one collaborator can be there in a cluster. Inter cluster links have maximum probability of congestion as they act as intermediate for the communication between two clusters, but since the basic idea behind the proposed approach is that there is least communication between the cores of different clusters, so chances of traffic congestion are very low. In the worst case if still there is congestion, than higher bandwidth of the inter cluster links will tackle this problem.

While designing the topology using the proposed approach, there are certain parameters which has to be pre-decided or pre-
calculated by some assumptions or past experiences. These parameters include three types of threshold values:

- **Cluster_threshold**: The number of clusters, which has to be formed while designing the customized topology, is considered.
- **Core_num_threshold**: The total number of cores that can be accommodated in any cluster are decided. This threshold is different for each cluster and varies depending upon the topology used for the particular cluster.
- **Collaborator_threshold**: The maximum number of collaborators for the topology is decided, so that traffic congestion can be controlled.
- **Common_core_threshold**: The maximum number of cores that can be the part of more than one cluster are decided, so that less traffic on the links is found.

A. Steps in the Algorithm

**Step I**: Consider the mesh topology.

**Step II**: Decide the number of clusters to be formed and set it as cluster_threshold. Pre define the number of cores that can be accommodated in a cluster as core_num_threshold. On the basis of past evaluation determine the number of collaborators and define collaborator_threshold. Define the total number of common cores for different clusters as common_core_threshold.

**Step III**: Identify the most communicating cores in the topology.

**Step IV**: Generate the communication graph.

**Step V**: Distribute the cores in the clusters on the basis of the amount of communication among them.

**Step VI**: Define the topology to be used for each cluster.

**Step VI**: Form the heterogeneous and hybrid clustered topology.

**Step VI**: Map the generated topology over the initially provided mesh topology.

V. EXPERIMENTAL SETUP

The experimental setup considered for the simulation of the proposed approach is the 6X6 mesh topology. We emphasis on the reduction of the number of hop counts between the most communicating cores, such that the latency involved is reduced, and number of wires are reduced due to which the complexity of the network is reduced along with the cost. Also the geometrical appearance of the overall network should not be changed, such that the space complexity of the whole network in the proposed approach remain same as that of the mesh topology. Basically, this approach combines the benefits of the both homogeneous and heterogeneous topologies as each cluster is formed using either of the topologies, but if the overall topology proposed in the paper is considered, then it forms the heterogeneous topology.

Fig. 1 shows the initially provided 5X5 mesh topology. Determine the most communicating cores in the given mesh topology. The most communicating cores have same color for differentiation. Some cores in the Fig. 1 has two colors, it determines that it is common to two of the clusters. Generated communication graph is shown in the Fig. 2. In this graph, the dashed lines shows that there is less amount of communication among the cores connected to the link. Bold line shows the connection between the most communicating cores. Now determine the clusters to be formed. As from the Fig. 2 we have decided to form 4 clusters and has set the cluster_threshold value as 4. After this decide the topology for each cluster. We have considered 4 topologies as 2 mesh and 2 bus topologies. The beauty of the proposed approach is that we can customize it according to the requirements of the system.
Any of the topology in any of the cluster can be used as per the requirements. Fig. 3 shows the formation of the clusters and generation of the heterogeneous and hybrid clustered topology. This topology generated can be arranged in such a way that it can be mapped onto the initially given mesh topology as shown in Fig.4. This mapping is not physical, it is just a representation to show that HHCT will consume the same amount of area as the mesh topology.

VI. EXPERIMENTAL RESULTS

In this section, the experimental results obtained during the simulation of the proposed approach are compared to the results obtained for the mesh topology. Fig. 5 gives the details of the latency involved for passing the packets between the nodes is considered. The latency considered is the end-to-end latency. As already discussed earlier in section I, the network latency is the total time taken by a packet to go from source to destination and back to the source, so it can be considered as the end-to-end latency considered two times. So instead of showing the results graphically we can conclude that the network latency is double the end-to-end latency. From the Fig. 5 and Fig. 6, it is clear that the HHCT is better than the mesh topology in terms of the latency and response time. The next parameter being considered is the loss probability. Loss probability when considered during the simulation of the proposed approach produces same results in both mesh and HHCT approach, and is nearly equivalent to 0.098.
VII. CONCLUSION AND FUTURE WORK

The proposed approach discussed in the paper basically modify the mesh topology in such a way that the number of hops between the most communicating cores are reduced such that latency, link utilization, loss probability, sink bandwidth and energy consumption involved is also reduced. While doing this the geometry of the overall topology obtained should not be affected and hence the same amount of space is consumed by the generated topology as by the original mesh topology. The clustering of the cores in the proposed approach has improved the performance of NoC over Mesh topology to a great extent.

As the future work we are planning to compare the proposed approach with the other existing NoC topologies such as Torus, Fat tree, hypercube etc. Also the analytical modelling of the proposed approach will be covered as the future work of this research.

REFERENCES