Routing on Network-on-Chip Using Artificial Intelligence-based Algorithms

A network on a chip or Network-on-Chip (NoC) is a network-based communication subsystem on an integrated circuit (IC), most typically between modules in a System-on-Chip (SoC). The modules on the IC are typically semiconductor IP cores schematizing various functions of the computer system and are designed to be modular in the sense of network science. Network-on-Chip is a router-based packet switching network between SoC modules. These routers use routing algorithms. In general, routing algorithms can be divided into two categories of adaptive and non-adaptive algorithms. Artificial Intelligence-based (AI-based) Algorithms fall into the category of adaptive algorithms. In this article, we will compare the simulation results of some adaptive and non-adaptive algorithms to investigate the artificial intelligence effect on NoC routing.

We need an appropriate simulator to check the results of different routing algorithms on NoC. NoC has several simulators, but, in this article, we specifically need a simulator that we can run AI-based algorithms on it and also, we can implement new AI-based algorithms on it. In this project, we use NIRGAM, which has all of these capabilities. NIRGAM is a systemC-based discrete event, cycle-accurate simulator for research in Network-on-Chip. It provides substantial support to experiment with NoC design in terms of routing algorithms and applications on various topologies. NIRGAM is a collaborative research between the University of Southampton UK and the Malaviya National Institute of Technology, Jaipur India.

In this article, we will consider the XY algorithm as an example of a non-adaptive algorithm. This algorithm is one of the simplest non-adaptive routing algorithms. In this routing, each node or router of NoC is identified by its (x, y) coordinates (for a 2D mesh). According to this algorithm, the data packets will traverse in X-direction towards the destination column. After finding the destination column, the data packets will traverse in Y-direction to the destination node.

We also consider the Q-routing (QRT) algorithm as an example of an adaptive algorithm. This algorithm is one of the simplest AI-based algorithms. Q-routing takes advantage of one of the reinforcement learning aspects, which is Q-learning. Based on the delivery time of the packet to every node, it builds the routing table that is called Q-table. Q-tables store latency information in terms of Q-values. Whenever a node sends a packet to a neighbor node, these values are updated. Q-routing dynamically learns and uses these Q-values to make a routing decision. By using these Q-values, it Routes the packet to the destination with the least delivery time.

Both XY and Q-routing algorithms are already implemented in NIRGAM. We will compare the results of them by giving the same traffic. To investigate more on adaptive algorithms, we implement the Confidence-based Q-routing (CQR) algorithm as a new AI-based algorithm on NIRGAM. This algorithm is quite similar to Q-routing, except that instead of fixing the learning rate, we have a variable learning rate. In this algorithm, each node has its own learning rate. These learning rates are stored in a table similar to Q-table and are dynamically updated and used.

To simulate, we give the same input to all three algorithms. After simulating these three algorithms in different situations in terms of topology size and network traffic, we see that the more saturated the network, the better the Confidence-based Q-routing algorithms, then Q-routing, and then XY, but, if the network is not very saturated, this relationship is reversed. Confident-based Q-routing is worse than Q-routing, and Q-routing is worse than XY in terms of hardware size, cost, and power consumption. Besides, the simulation results show that Q-routing, Confidence-based Q-routing do not have a much worse or much better answer than each other.

In Table 1, as an example of simulations, one of the conditions in which these three algorithms are simulated is given. Table 2 shows the results of the simulation for these three algorithms.

Table 1: NOC Specification

Topology	5x5 Mesh
Clock Frequency	1 GHz
Routing Algorithms	XY, QRT, CQR
Simulation Number	5000 Clock Cycles
Switching	Wormhole

Table 2: Simulation Results and Comparison

Routing	Average Latency (Cycles/flit)	Power (w)
XY	162	0.058
QRT	127.76	0.077
CQR	100.74	0.084

The reason for these results can be the different degree of complexity of these three algorithms and the difference in their compatibility; therefore, the routing algorithm of the system should be chosen, depending on the application in which the system is going to be used.