



# The odd-even turn model for adaptive routing with node failure scenario

<sup>1</sup>N.Zahira Jahan MCA.,M.Phil., Associate Professor,

<sup>2</sup>Ms P.Umarani Final MCA,

Department Of MCA, Nandha Engineering College (Autonomous), Erode-52.

E-Mail ID: zahirajahan1977@gmail.com, Umarani3495@gmail.com

*Abstract-The mesh topology has become a popular interconnection architecture for constructing massively parallel multi-processors. In particular, low-dimensional meshes have been used in several academic and commercial machines, such as the DASH multiprocessor. This project presents a model for designing adaptive wormhole routing algorithms for meshes. The model restricts the locations where some turns can be taken so that deadlock is avoided. In comparison with previous methods (xy routing method), the degree of routing adaptiveness provided by the model is more even for different source-destination pairs. The wireless mesh network may benefit from this feature in terms of communication efficiency. The odd-even turn model makes message routing very minimal. In addition, this property results in a smaller fluctuation of the network performance with respect to different traffic patterns. In addition, the node failure scenario is also considered so that if the path contains that node, then new alternate path from the failure node to the destination is recalculated or taken from cache if previous path information is stored.*

## 1 INTRODUCTION

The mesh topology has become a popular interconnection architecture for constructing massively parallel multi-processors. In particular, low-dimensional meshes have been used in several academic and commercial machines, such as the DASH multiprocessor [7], Intel TFLOPS supercomputer [1], and Intel Paragon [6]. Processors (or nodes) of a mesh communicate with each other by sending messages through the underlying network. Hence, efficient communication is critical to the performance of a mesh. Recently, the most popular technique for switching packets is wormhole routing [2]. With wormhole routing, a packet is divided into flow control digits (or flits). The flits are routed through

the network one after another in a pipeline fashion. The first flit of a packet is designated as the header flit, which contains routing information and leads the packet through the network. When the header flit is blocked from advancing due to lack of output channels, all of the flits wait at their current nodes for available channels. Each router only requires small buffer space to store the flits and communication latencies are low with wormhole routing. Routing algorithms are used to determine the sequence of channels a message packet traverses from the source to the destination. A desirable property of a routing algorithm is freedom from deadlock and livelock. Livelock occurs when a message proceeds through the network indefinitely, never arriving at its destination. Livelock is possible only if message routing is adaptive and is nonminimal. Deadlock is caused by packets waiting for each other in a cycle. In wormhole routing, the order in which channels are used for a packet must meet certain criteria so that deadlock is prevented. Adaptiveness is also an important factor for message routing [6].

Adaptiveness increases the chances that packets may avoid hot spots or faulty components and reduces the chances that packets are continuously blocked. There have been many routing algorithms proposed for meshes that are based on wormhole routing in the literature [2], [3], [5]. Routing algorithms can be generally classified into three categories, depending on the degree of adaptiveness provided by the algorithms. A nonadaptive routing algorithm is deterministic and routes a packet from the source to the destination along a unique, predetermined path. A minimal fully adaptive routing algorithm routes all packets through any shortest paths to the destinations.

A partially adaptive routing algorithm allows multiple choices for routing packets via shortest paths, but it does not allow all packets to use any shortest paths. In [2], [3], virtual channels were

introduced to assist the design of nonadaptive routing algorithms so that deadlock is avoided. Virtual channels are abstractions that share the same physical channel. Later, several researchers [3], [5] used virtual channels to design partially adaptive and fully adaptive routing algorithms for a variety of network architectures, including meshes. Adding virtual channels allows the design of highly adaptive routing algorithms. In fact, it is impossible to produce a deadlock-free fully adaptive routing algorithm for a mesh without addition of virtual channels. However, adding virtual channels to meshes is not free. It involves adding buffer space and complex control logic to routers, thus communication performance of the network and reliability of the routers may be affected [4]. Furthermore, deadlock-free routing schemes that are not based on adding virtual channels may be used as the basic mechanisms for implementing adaptive routing algorithms that use virtual channels [1]. Hence, the development of deadlock-free routing algorithms that do not use virtual channels is important. Several routing algorithms that require no virtual channels have been proposed for mesh networks.

The xy routing algorithm [7] for two-dimensional meshes routes a packet first along x dimension (dimension 0) and then along y dimension (dimension 1). The xy algorithm ensures deadlock freedom, but it provides no adaptiveness. Glass and Ni [5], [6] presented an elegant technique, called turn model, for designing partially adaptive wormhole routing algorithms that require no virtual channels. The basic idea of the model is to prohibit the minimum number of turns that break all of the cycles so that deadlock can be avoided. Based on the turn model, three partially adaptive routing algorithms, namely west-first, north-last, and negative-first, were presented for two-dimensional meshes. Unfortunately, the degree of adaptiveness provided by the turn model is highly uneven. Using any of the above-mentioned routing algorithms, at least half of the source-destination pairs are restricted to having only one minimal path, while full adaptiveness is provided for the rest of the pairs. Such uneven adaptiveness not only causes unfairness but also curtails the ability of the model in alleviating traffic congestion problem. Performance of the network communication may be affected as a result.

The xy algorithm can also be considered as achieving deadlock freedom by prohibiting certain turns. In [2], a model called direction restriction model was proposed to facilitate the design of partially adaptive routing algorithms. This model is based on dividing a system into two unidirectional networks. Message routing is done in two phases. In the first phase, a packet is routed adaptively to an intermediate node using one unidirectional network. It is then routed adaptively to the

destination using the other network in the second phase. The degree of adaptiveness provided by the model is also highly uneven.

In this paper, we propose a novel model for designing partially adaptive, deadlock-free routing algorithms for meshes. The model is not based on adding virtual channels to network topologies. Unlike the previous methods, which rely on prohibiting certain turns in order to achieve deadlock freedom, our model restricts the locations where some types of turns can be taken. The degree of routing adaptiveness provided by the proposed model is more even for different source-destination pairs. The mesh network may benefit from this feature in terms of communication efficiency. Extensive simulations have been conducted to evaluate the performance of the model.

The simulation results show that the even adaptiveness provided by our odd-even turn model is important under nonuniform traffic. It makes message routing less vulnerable to nonuniform factors, such as hot spot traffic. Furthermore, this property allows the performance of the network to have smaller fluctuation with respect to different types of traffic. In the next section, the odd-even turn model for two-dimensional meshes is presented. Simulation results for two-dimensional meshes are shown in Section 3. Section 4 addresses an extension of the model to higher-dimensional meshes. Section 5 draws the conclusions.

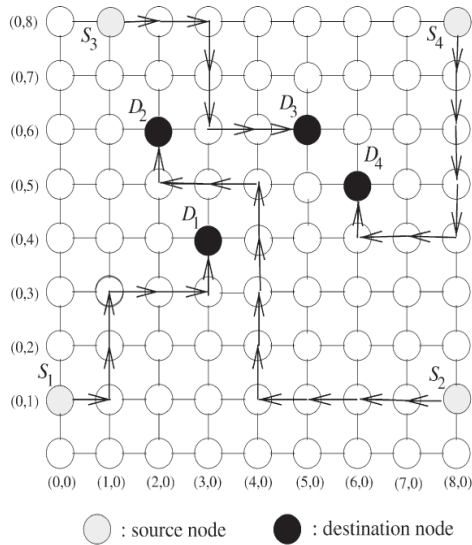
## 2 ROUTING IN TWO-DIMENSIONAL (2D) MESHES

In this section, we present a model, called the odd-even turn model, to facilitate deadlock-free routing in two-dimensional (2D) meshes with no virtual channels. In a  $K_0 \times K_1$  2D mesh, a node  $X$  is identified by a two-element vector  $(x_0; x_1)$ ,  $0 \leq x_0 \leq K_0 - 1$  and  $0 \leq x_1 \leq K_1 - 1$ , where  $x_0$  and  $x_1$  are called the coordinates of dimension 0 and dimension 1 of  $X$ , respectively. To facilitate the presentation, we label the four sides of a 2D mesh as East, West, South, and North. All of the nodes that have the same coordinates of dimension 0 constitute a column, and all of the nodes that have the same coordinates of dimension 1 constitute a row. Row channels refer to channels along dimension 0; that is, a row channel connects two neighboring nodes on the same row.

Similarly, column channels refer to channels along dimension 1. Further, a column channel is called an SN (respectively, NS) channel if its direction is from South to North (respectively, North to South). Suppose that a channel links node  $A$  to node  $B$ . Nodes  $A$  and  $B$  are called the tail node and the head node of the channel, respectively.

**Definition 1.** A turn consists of a row channel and a column channel such that the tail node of one of the channels is the head node of the other; the

common node of the two channels is called the turning node of the turn. Essentially, a turn involves a 90-degree change of traveling direction. Unless otherwise specified, a turn is a 90-degree turn in the following description. There are eight types of turns, according to the traveling directions of the associated channels. A turn is called an ES turn if it involves a change of direction from East to South. Similarly, we can define the other seven types of turns, namely EN, WS, WN, SE, SW, NE, and NW turns, where E, W, S, and N indicate East, West, South, and North, respectively. A packet is said to take a turn at node X if it traverses the turn and the turning node of the turn is X.



**2.1 The Odd-Even Turn Model**

Deadlock in wormhole routing is caused by packets waiting on each other in a cycle. Previous methods, such as the turn model [6] and the xy algorithm, avoid deadlock by prohibiting certain turns. Instead, the odd-even turn model is based on restricting the locations at which certain turns can be taken so that a circular wait can never occur. The proposed model does not eliminate any types of turns for message routing.

Definition 2. In a 2D mesh, a column is called an even (respectively, odd) column if the dimension-0 coordinate of the column is an even (respectively, odd) number.

The basic idea of the odd-even turn model is to restrict the locations where some of the turns can occur so that an EN turn and an NW turn are not taken at nodes in the same column, and neither are an ES turn and an SW turn. As shown later, this may avoid deadlock. More precisely, the odd-even turn model is governed by the following two rules:

Rule 1. Any packet is not allowed to take an EN turn at any nodes located in an even column, and it is not allowed to take an NW turn at any nodes located in an odd column.

```

Algorithm ROUTE
/* Source node: (s0, s1); destination node: (d0, d1); current node: (c0, c1). */
begin
  Avail_Dimension_Set ← ∅;
  c0 ← d0 - c0;
  c1 ← d1 - c1;
  if (c0 = 0 and c1 = 0)
    Deliver the packet to the local node and exit;
  if (c0 = 0) /* currently in the same column as destination */
    if (c1 > 0)
      Add North to Avail_Dimension_Set;
    else
      Add South to Avail_Dimension_Set;
  else
    if (c0 > 0) /* eastbound messages */
      if (c1 = 0)
        Add East to Avail_Dimension_Set;
      else {
        if (c0 is odd or c0 = s0)
          if (c1 > 0)
            Add North to Avail_Dimension_Set;
          else
            Add South to Avail_Dimension_Set;
        if (d0 is odd or c0 ≠ 1) /* odd destination column or ≥ 2 columns to destination */
          Add East to Avail_Dimension_Set;
      }
    else { /* westbound messages */
      Add West to Avail_Dimension_Set;
      if (c0 is even)
        if (c1 > 0)
          Add North to Avail_Dimension_Set;
        else
          Add South to Avail_Dimension_Set;
    }
  }
  Select a dimension from Avail_Dimension_Set to forward the packet;
end
    
```

Rule 2. Any packet is not allowed to take an ES turn at any nodes located in an even column, and it is not allowed to take an SW turn at any nodes located in an odd column.

**2.2 Degree of Adaptiveness**

$$P_{\text{fully adaptive algorithm}} = \frac{(d_x + d_y)!}{d_x!d_y!}$$

$$P_{\text{odd-even turn model}} = \begin{cases} \frac{(d_x+h)!}{d_x!h!} & \text{if column } x_s \text{ is an allowable} \\ & \text{column and } d_x \text{ is an odd} \\ & \text{number,} \\ \frac{(d_x+h)!}{d_x!h!} & \text{otherwise.} \end{cases}$$

In case the packet is an NW or SW packet, i.e.,  $x_x = 0$ , the degree of adaptiveness for the odd-even turn model becomes

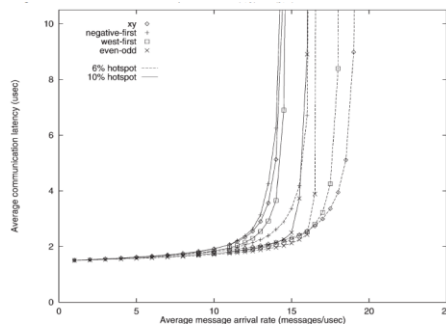
$$P_{\text{west-first}} = \begin{cases} \frac{(d_x+d_y)!}{d_x!d_y!} & \text{if } x_d \geq x_s \\ 1 & \text{otherwise.} \end{cases}$$

$$P_{\text{negative-first}} = \begin{cases} \frac{(d_x+d_y)!}{d_x!d_y!} & \text{if } (x_d \leq x_s \text{ and } y_d \leq y_s) \text{ or} \\ & (x_d \geq x_s \text{ and } y_d \geq y_s) \\ 1 & \text{otherwise} \end{cases}$$

**3 PERFORMANCE EVALUATION**

To evaluate the performance of the odd-even turn model, we have developed an event-driven simulator. The simulations were conducted on a 15 × 15 mesh under various traffic patterns. Two unidirectional channels exist between each pair of neighboring nodes. All of the channels have the same bandwidth of 20 flits/sec. Each input channel has a buffer the size of a single flit. We compare our algorithm with the nonadaptive xy

algorithm and the west-first and negative-first algorithms that are based on the turn model [1]. In the simulation, we consider minimal routing of messages. For the odd-even turn model, algorithm ROUTE is used. When multiple header flits wait for the same available output channel, the local first-come-first-served policy [6], which decides in favor of the header flits that arrive at the node first, is adopted. When a header flit has two output channels available for it, our algorithm uses the output channel along dimension 1.



#### 4 EXTENSION TO HIGH-DIMENSIONAL MESHES

In this section, we show that the odd-even turn model can be readily extended to high-dimensional meshes. To facilitate the presentation, we will address 3D meshes. Let the three dimensions of a 3D mesh be denoted by  $x$ ,  $y$ , and  $z$ . Further, the two directions associated with  $x$  dimension are represented by  $x$  and  $x\bar{y}$ , which correspond to directions with increasing and decreasing  $x$  coordinate, respectively. Similarly, we can define directions  $y$ ,  $y\bar{z}$ ,  $z$ , and  $z\bar{x}$ . In a 3D mesh, there are 24 90-degree turns. We use  $x.y$  to denote a turn that involves a change of direction from  $x$  to  $y$ . The notations for the other types of turns are used in a similar fashion.

A  $yz$ -plane consists of all the nodes that have the same coordinates of dimension  $x$ . An even (respectively, odd)  $yz$ -plane is a  $yz$ -plane whose dimension- $x$  coordinate is an even (respectively, odd) number. To facilitate the description of our scheme, the following four turns  $x.y$ ,  $x.y\bar{z}$ ,  $x.z$ , and  $x.z\bar{y}$  collectively constitute the class of  $x.\bar{y}$   $yz$  turns. Further, the turns  $y.x\bar{y}$ ,  $y\bar{x}\bar{y}$ ,  $z.x\bar{y}$ , and  $z\bar{x}\bar{y}$  collectively constitute the class of  $yz.\bar{y}$   $x$  turns. In a 3D mesh, the odd-even turn model is governed by the following rules:  $yz$  Rule. On any  $yz$ -plane, Rule 1 and Rule 2 adopted for 2D mesh routing are used, with dimension  $y$  corresponding to dimension 0 and dimension  $z$  corresponding to dimension 1.  $x$  Rule. A packet is not allowed to take any of the  $x.\bar{y}$   $yz$  turns at a node located in an even  $yz$ -plane, and it is not allowed to take any of the  $yz.\bar{y}$   $x$  turns at a node located in an odd  $yz$ -plane.

Essentially, we do not allow an  $x.\bar{y}$   $yz$  turn to be connected to a  $yz.\bar{y}$   $x$  turn so that the property of deadlock freedom can be achieved. The existence

of this property can be readily argued by following the proof of Theorem 1.

#### 5 CONCLUSIONS

We have presented the odd-even turn model for designing partially adaptive wormhole routing algorithms without adding virtual channels. In comparison with the well-known turn model [7], our scheme provides more even routing adaptiveness. Simulation results demonstrate that communication performance of the meshes may be improved under nonuniform traffics using the proposed model.

Although fault tolerance is not addressed in this paper, the odd-even turn model may be useful for designing fault-tolerant routing algorithms. The relative locations where restrictions on the individual turns are placed is an issue that requires further study in this respect. In addition, nonminimal routing and 180 degree turns must also be considered to avoid faulty components. This is an interesting subject for future work.

#### REFERENCES

- [1] R.V. Boppana and S. Chalasani, "A Comparison of Adaptive Wormhole Routing Algorithms," Proc. Int'l Symp. Computer Architecture, pp. 351±360, May 1993.
- [2] Y.M. Boura and C.R. Das, "A Class of Partially Adaptive Routing Algorithms for  $n$ -Dimensional Meshes," Proc. 1993 Int'l Conf. Parallel Processing, pp. III±175±III±182, 1993.
- [3] Y.M. Boura and C.R. Das, "Efficient Fully Adaptive Wormhole Routing in  $n$ -Dimensional Meshes," Proc. Int'l Conf. Distributed Computing Systems, pp. 589±596, 1994.
- [4] A.A. Chien "A Cost and Speed Model for  $k$ -Ary  $n$ -Cube Worm-hole Routers," Proc. Hot Interconnects '93, Aug. 1993.
- [5] A.A. Chien and J.H. Kim, "Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors," J. ACM, vol. 42, no. 1, pp. 91±123, Jan. 1995.
- [6] Intel Corporation, Paragon XP/S Product Overview, 1991.
- [7] Intel Corporation, A Touchstone DELTA System Description, 1991.
- [8] W.J. Dally, "Virtual-Channel Flow Control," IEEE Trans. Parallel and Distributed Systems, vol. 3, no. 2, pp. 194±205, Mar. 1992.