

Adaptive Routing Algorithms and Implementations

David Ouellet-Poulin

School of Information Technology and Engineering (SITE)

University of Ottawa, Ottawa, Ontario

Email: douel025@site.uottawa.ca

Abstract—Numerous adaptive routing algorithms which are common in traditional networking are being adapted for use in inter-processor communications. Although this method has numerous advantages, the overhead in terms of speed and node size has thus far been prohibitive in the approach's success. Future system-on-chips as well as network-on-chips may implement a form of such a routing algorithm but not as a central feature.

I. INTRODUCTION

The issue of interprocessor communication has become paramount due to the ever-increasing speed of each unit and the amount of Processing Elements (PEs) now included in mainstream systems. Various architectures and routing algorithms have appeared in the last two decades in order to decrease the overhead to both the data rate as well as chip size. This document contains an overview of popular tactics devised for adaptive routing on multiprocessor microchips.

II. PRIOR ART

Adaptive routing proposes that each router be aware of the network's traffic situation and adapt its routing (worm-hole packet switching) accordingly. The issue is to avoid traffic congestions and be fault-tolerant towards both disabled nodes and connection [1]. Therefore, certain algorithms permit misrouting which leads packets away from their intended destination to avoid high-traffic areas. In the case where traffic is low, an adaptive routing algorithm should seek to provide a minimal (ideally the shortest) path between source and destination. Implementations of adaptive routing can cause adverse effects if care is not taken in analyzing the behavior of the algorithm under different scenarios (concentrated traffic, non-uniform and uniform traffic). The following sections will discuss the following problems and metrics that are common to all routing algorithms: cyclical resource dependence (deadlocks), starvation and the adaptiveness of a given approach.



A. Deadlocks

Deadlocks in fully-adaptive routing can be a very difficult problem to solve since there are an infinite amount of possible traffic scenarios. Node buffer size as well as topology will dictate the probability of resource deadlock. All solutions discussed in this document include a form of flow restrictions, which is to say that only certain abstract "turns" along the topology are allowed.

B. Livelocks

A livelock is a type of starvation that can occur in adaptive routing where misrouting is permitted. Packets that were routed away from their destination may become locked in a loop sequence that persistently re-routes them away from their goal due to local congestion. The concept of *fairness* must be included in an algorithm in order to preclude this situation. Other solutions involve the same type of restrictions that are used for preventing deadlocks. For instance, the Turn Model restricts turns that can form cycles while Planar-Adaptive uses minimal routing (does not permit misrouting) [2, 3].

C. Degree of Adaptiveness

The metric used to gauge the effectiveness of an adaptive routing algorithm is the most important metric in comparing the different approaches. It is the number of shortest paths the algorithm allows from source node to destination node. The canonical benchmark, a fully adaptive algorithm's degree of adaptiveness (for a 2d-mesh) from source node (s_x, s_y) to destination node (d_x, d_y) , is the following [2]:

$$S_f = \frac{(\Delta x + \Delta y)!}{\Delta x! \Delta y!} \quad (1)$$

Where $\Delta x = |d_x - s_x|$ and $\Delta y = |d_y - s_y|$. While formula 1 is specific to 2d-meshes, it can be determined for other topologies by determining the number of permutations possible while retaining minimal path length.

D. Partially Adaptive vs. Fully adaptive

Fully adaptive routing can route every packet along any of the shortest paths in the topology, while partially adaptive routing cannot. Thus from the degree of adaptiveness defined in the previous section, we may think of a partially adaptive routing algorithm as the following:

$$\frac{S_p}{S_f} \leq 1 \quad (2)$$

where S_p is the degree of adaptiveness of the partially adaptive routing algorithm in question (S_f is defined in for 1).

III. ALGORITHMS

A great variety of adaptive routing algorithms have been devised for networking in the more traditional sense. However, adaptive algorithms for routing in computer systems with multiple PEs on-chip are more recent. They usually work by introducing flow control techniques that provide the adaptive behavior while precluding the possibility of deadlocks [4]. Each approach may or may not be limited to a specific topology; both options are explored here.

A. Turn Model

The turn model is an approach which is used for designing wormhole routing algorithms that are deadlock free, livelock free, minimal or nonminimal and maximally adaptive and does not require additional channels (physical or virtual). The model analyzes the directions of turns in a network and the cycles that these turns can form. This works for any k -ary n -cubes which makes it a very powerful model albeit not a fully adaptive one [2].

For a 2d-mesh, deadlocks occur when packets waiting for each other form a cycle. All channels are separated into sets, one for each *virtual* direction, after which all possible turns from one direction to another are determined (180-degree and 0-degree are ignored). Subsequently, all the cycles that can be formed from these turns are generated; from each of these, one type of turn must be prohibited in order to preclude the possibility of deadlocks and livelocks. As demonstrated in Fig. 1.

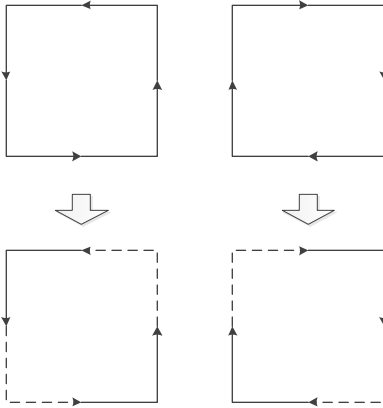


Figure 1. Possible abstract cycles in 2D mesh turn modeling. Dashed lines are prohibited turns [2].

Routing of packets must take place only using the sets of turns that have been created from the topology analysis. The resulting algorithm is not minimal as it prohibits turns that could lead to the shortest path.

The degree of adaptiveness of algorithms created using this model can reach fully adaptive in ideal cases but will generally lie below 72%. Simulations of such algorithms for 2d-meshes have determined that the average communication latency is exponential to the network throughput. They also confirm that adaptive routing performs much better than deterministic or oblivious for non-uniform traffic.

B. Odd-Even Turn Model

An improvement to the turn model for meshes is to restrict turns only in certain locations of the topology. The odd-even turn model stems from restricting certain turns depending on the odd-ness or even-ness of the column the packet is in. This simple modification allows for a greater number of possible paths while remaining deadlock and livelock free [5]. The degree of adaptiveness is the following:

$$P_{\text{odd-even}} = \frac{(d_y + h')!}{d_y!h'!} \quad (3)$$

or

$$P_{\text{odd-even}} = \frac{(d_y + h)!}{d_y!h!} \quad (4)$$

Where $h = \lceil \frac{d_x}{2} \rceil$ and $h' = \lceil \frac{d_x-1}{2} \rceil$. Depending on the odd-ness or even-ness of the column in question. Clearly, this is a more adaptive algorithm than the standard turn model.

C. Planar

For the case where the topology in question contains a large number of dimensions, the low-cost alternative is the planar-adaptive approach. The basic idea is to limit routing to a two dimensions (routing planes) at a time as seen in Fig. 2. The packets travel through a set of planes until they arrive to their destination. It is important to note that the routing along each plane is not adaptive (packets may use any path). This is necessary to limit the resources needed for the routing procedure.

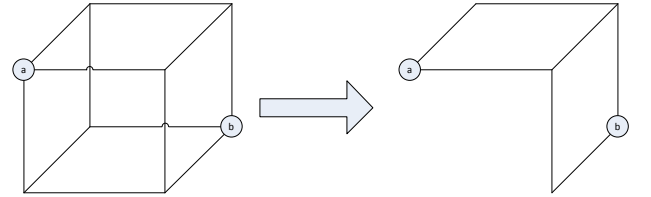


Figure 2. Graphical demonstration of limiting dimensions for planar-adaptive routing of a cube [3].

This approach eliminates the need for a large number of virtual (additional) channels while drastically reducing the chance of deadlocks (deadlock free if fault free). In fact, planar-adaptive only needs a constant number of virtual channels regardless of the number of dimensions. For fault-tolerance, the addition of misrouting completely eliminates deadlocks while limiting livelocks to a very small probability.

In addition, implementations of this algorithm are fairly straightforward and require very little logic. Simulations have shown that planar-adaptive outperforms deterministic routing while using the same amount of resources. The addition of more virtual channels can lead to even higher performance [3].

D. GOAL

The Globally Oblivious Adaptive Locally or GOAL for Torus Networks is an approach which complements the planar-adaptive method. Here, the routing is adaptive on the current dimension while the switch from one dimension to the next is performed randomly (obliviously). This allows for a balanced load on channels connecting dimensions and on each dimensional plane. Once a dimension has been chosen the packet travels in a minimal direction towards its intended destination. However, the initial direction (since a torus wraps around) is chosen as to balance the load on the dimension [6].

The 2-dimensional plane on which the packet is located is divided into the four quadrants of the cartesian plane (with the current location placed at the origin). Each possible direction is weighted accordingly to the shortness of the resulting path. The final direction is picked via a probability function based on these weights. This allows for greater usage of resources in the case of non-uniform traffic while keeping paths fairly minimal in more uniform cases.

Similarly to the planar approach, GOAL can become deadlock free with the addition of three virtual channels per unidirectional physical channel. In addition, GOAL is livelock free because of the nature of the torus topology and the manner in which dimensional routing is performed.

IV. EXAMPLE SYSTEMS

Although multi-processor systems are becoming the norm in mainstream desktops and laptops, the number of processor elements is not yet high enough to warrant any type of routing. Therefore most of the implementations reviewed here are from prototypes or non mass-market products.

A. IBM Cell

The cell processor architecture depends on deterministic routing due to its simple topology and small number of nodes. The eight processors are connected in a ring topology (four connection wide) with an arbiter allocating transfers and ensuring routes do not proceed more than half-way (4 hops) around the ring [7].

B. Intel TeraFLOPS

The research prototype processor featuring 80 cores features a flexible routing strategy: deterministic, oblivious and adaptive algorithms are supported. Each node contains a 5-port message passing router and are connected in an 8x10 2d-mesh [8].

C. Tiler TILE64

As the name suggests, this system uses 64 processing units which are connected in a 4-dimensional crossbar mesh structure of 4 nodes each with each node having its own L1 and L2 cache. The routing is deterministic and circuit switching is used. Since the chip is oriented for real-time processing, the overhead of adaptive routing makes it a prohibitive choice [9].

D. STMicroelectronics STNoC

The STNoC is a network-on-chip processor with 6 PEs which uses a hybrid topology of ring and point-to-point called Spidergon (Fig. 3). The algorithm used is Across-First routing which is a deterministic source-routing approach that does not prevent deadlocks [4].

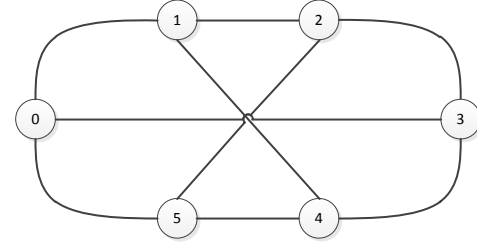


Figure 3. Spidergon topology of the STMicroelectronics STNoC chip [4].

V. CONCLUSION

There is a great variety of adaptive routing algorithms in the literature but few actual implementations in products. This is due to the nature of adaptive routing, which constantly re-thinks the path packets are following as it makes its way across the network. Thus introducing overhead and needing additional connections (virtual channels) and increasing the complexity of each router, thus augmenting the amount of logical elements and size (on the die) necessary. Since there are no systems available to the consumer that contain more than eight cores (IBM Cell), it is understandable that routing for system-on-chips remains very basic in order to minimize communication latency between each PE on a chip.

Since most algorithms are deadlock and livelock free, the most important metrics for adaptive routing algorithm lie in the complexity of each router, the addition of virtual channels and the latency of communication. Also of importance is the degree of adaptivity, which help keep the network running smoothly under non-uniform traffic. It is interesting to note however that few publications care to calculate their degree of adaptiveness.

However, as MOSFET-based electronics begin to reach the performance wall in terms of clock rate, mainstream computing products should begin to see a steep increase in processing elements. As the number of nodes climb, deterministic and oblivious routing cannot scale to meet the demand. Therefore, it is not illogical to expect that adaptive routing algorithms become used in mainstream system-on-chip as well as network-on-chip systems.

REFERENCES

- [1] W. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 4, pp. 466–475, Apr. 1993.
- [2] C. Glass and L. Ni, "The turn model for adaptive routing," in *Computer Architecture, 1992. Proceedings., The 19th Annual International Symposium on*, 1992.

- [3] A. Chien and J. H. Kim, "Planar-adaptive routing: Low-cost adaptive networks for multiprocessors," in *Computer Architecture, 1992. Proceedings., The 19th Annual International Symposium on*, 1992.
- [4] N. E. Jerger and L.-S. Peh, "On-chip networks," *Synthesis Lectures on Computer Architecture*, vol. 4, no. 1, pp. 1–141, 2009.
- [5] G.-M. Chiu, "The odd-even turn model for adaptive routing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, pp. 729 –738, July 2000.
- [6] A. Singh, W. Dally, A. Gupta, and B. Towles, "Goal: a load-balanced adaptive routing algorithm for torus networks," in *Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on*, pp. 194 – 205, 2003.
- [7] M. Gschwind, H. Hofstee, B. Flachs, M. Hopkin, Y. Watanabe, and T. Yamazaki, "Synergistic processing in cell's multicore architecture," *Micro, IEEE*, vol. 26, no. 2, pp. 10 –24, 2006.
- [8] I. Corporation, "Intel teraflops research chip overview," 2007.
- [9] T. Corporation, "Tile64 processor overview," 2009.