Centralized Adaptive Routing for NoCs

Ran Manevich[1], Israel Cidon[2], Avinoam Kolodny[2] and, Isask'har Walter[1]
Electrical Engineering Department, Technion – Israel Institute of Technology, Israel
[1]{ranman,zigi}@tx.technion.ac.il [2]{cidon,kolodny}@ee.technion.ac.il

Abstract— As the number of applications and programmable units in CMPs and MPSoCs increases, the Network-on-Chip (NoC) encounters diverse and time dependent traffic loads. This trend motivates the introduction of NoC load-balanced, adaptive routing mechanisms that achieve higher throughput as compared with traditional oblivious routing schemes that are perceived better suited for hardware implementations. However, an efficient adaptive routing scheme should base its decisions on the global state of the system rather than on local or regional congestion signals as is common in current adaptive routing schemes. In this paper we introduce a novel paradigm of NoC centralized adaptive routing, and a specific design for mesh topology. Our scheme continuously monitors the global traffic load in the network and modifies the routing of packets to improve load balancing accordingly. In our specific mesh-based design, XY or YX routes are adaptively selected for each source-destination pair. We show that while our implementation is scalable and lightweight in hardware costs, it outperforms distributed adaptive routing schemes in terms of load balancing and throughput.

Index Terms— Network-on-Chip, routing algorithms, adaptive routing, load balancing.

1 Introduction

Network-on-Chip (NoC) routing policy is among the most important considerations in on-chip networks design. In general, routing algorithms can be classified as oblivious or adaptive. In oblivious routing, paths are determined solely by the source and destination addresses. Oblivious routing algorithms, such as Dimension Ordered Routing (DOR), are typically selected for NoCs since they are efficiently implemented in hardware, simple to test and are deadlock free. The main drawback of these algorithms is their poor load balancing that results in low throughput under high load and diverse traffic conditions. In adaptive routing, paths are dynamically adjusted to the changing traffic patterns to avoid local congestion and achieve a more uniform utilization of links. Although adaptive routing adds considerable design and testing complications, it is addressed in the NoC literature due to its potential performance benefits.

Preferably, adaptive routing should take into account the global traffic load of the network (i.e. centralized adaptive routing). This statement appears in almost every work that deals with adaptive routing (e.g. [3], [4], [7]). However, the same papers claim that centralized adaptive routing is difficult to implement and introduce designs where routers make decisions based on local [4] or regional [3], [7] loads using local or neighbors information. As a result of such short-sighted local decisions, these routers might direct traffic towards congested areas.

In this paper we present the architecture of a novel centralized adaptive routing mechanism that makes routing decisions based on the global state of the traffic in the NoC. Our architecture is comprised of an off-network congestion aggregation logic and a central routing controller. As a first realization for a 2D-mesh NoC, we introduce Adaptive Toggle Dimension Order Routing (ATDOR). In ATDOR, for every source-destination pair, paths are adaptively switched between XY and YX. The idea to split the traffic to XY and YX routes was first introduced in random oblivious routing algorithms [6] as O1TURN and in [2] as TXY.

We evaluate the hardware cost of ATDOR and compare its performance with both oblivious and distributed adaptive routings. We show that ATDOR outperforms the latter schemes in load balancing and throughput for various traffic patterns and NoC sizes. Moreover, we show that ATDOR demands very few extra hardware resources and scales well even for very large systems.

The paper is organized as follows: In section 2, we present the concept and hardware architecture of our method. Simulation results are presented in section 3. Section 4 concludes the paper.

2 Concepts and Architecture

The centralized adaptive NoC routing mechanism is composed of two modules, a feedback module and a control module. The feedback module monitors the traffic load across the network and aggregates it into a central traffic load map. The control module adjusts the routing according to the most updated traffic load map, in order to maximize load balancing and reduce congestion.

2.1 Feedback

The feedback module is composed of three functional layers. The first layer is implemented at the routers and is responsible for monitoring the local in-router load. There are several common metrics used to measure this load, such as the utilization of buffers, VCs (virtual channels), or crossbar. We assume the implementation of this layer follows one of the previously proposed local load measurements such as in [3]. Each router includes a K-bits register that indicates the nodal load. The second layer is

Manuscript submitted: 25-Aug-2010. Manuscript accepted: 28-Sep-2010.
Final manuscript received: 08-Oct-2010.

Posted to the IEEE & CSDL on 11/3/2010
DOI 10.1109/L-CA.2010.17 1556-6056/10/$26.00 © 2010 Published by the IEEE Computer Society

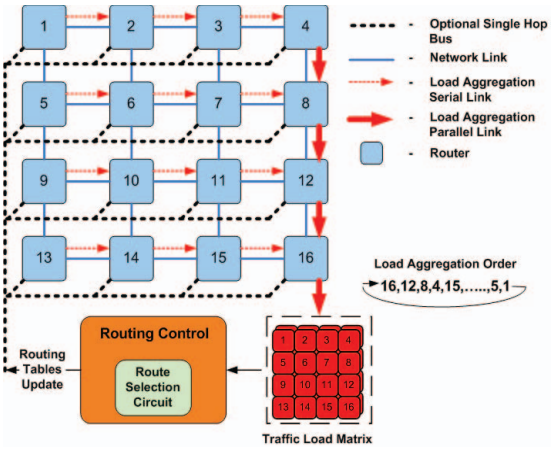


Fig. 1. Architecture of a 4x4 2D mesh.

responsible for aggregating the local load measurements into a traffic load map. We use a dedicated aggregation infrastructure aside the NoC, as depicted in Figure 1, to form an efficient, predictive and high speed feedback loop. Load values are sequentially moved towards the right column through serial links (thin arrows), and then down to the traffic load matrix through parallel links (thick arrows). The throughput of this implementation is K bits per clock cycle and its overhead is one additional wire per NoC link at most. The third layer is the Traffic Load Matrix (TLM). The TLM contains scalar local load measurements for each router reflecting the most updated overall traffic congestion information. The TLM is implemented with double buffering: while the local load measurements are collected and updated in the first TLM buffer, the routing calculation is conducted with the values previously stored in the second buffer. Hence, the load aggregation process works in iterations where in each iteration all K bits values from all routers are brought to the first buffer and copied in one clock cycle to the second buffer.

2.2 Routing Control

Our architecture is based on source routing where routing tables are located at the source modules and the information that determines the whole path is carried within the first flit of each packet. The Routing Control Module (RCM) is responsible for calculating the routing tables in the network according to the contents of the TLM. Routing between source destination pairs are dynamically adjusted to improve load balancing and avoid congestion. Following each update iteration of the TLM, the RCM examines a set of source-destination pairs and computes the updated least congested route for each pair. Once the RCM completes the evaluation of all the destinations of a particular source, it updates the corresponding source routing table. The sequence of flows that are rerouted, the timing between routing adjustments and the update rate of the TLM are determined by the control unit. Conceptually, the control unit can be a simple state machine that periodically scans all the source-destination pairs or a sophisticated microcontroller that implements dynamic prioritization among the active flows. In most cases the update of the routing tables can be performed through the NoC itself. However, in systems with highly fluctuating traffic, to achieve a fast feedback loop, we propose to utilize prioritization [1] or to enhance the network with a low latency signaling bus [5] for the routing tables update.

Finding the least congested route is a complex real time task even if we confine ourselves to shortest path routes. Moreover, the use of such routes requires the source routing tables to include detailed hop-by-hop path information and the routers to perform a complex deadlock free mechanism. In order to simplify the overall solution and reduce hardware costs, we define a new routing architecture termed Adaptive Toggle Dimension Order Routing (ATDOR). In ATDOR paths are adaptively toggled between XY and YX routing for every source-destination pair individually. This allows using one bit wide source routing tables at the sources. Moreover, only two VCs are needed (one for the XY routes and the other for YX routes) in order to avoid deadlocks. Finally, it allows the implementation of a fast and simple route selection circuit at the RCM. The architecture of a combinational pipelined route selection circuit is presented in Figure 2. The proposed circuit receives the IDs of source and destination modules from the routing control wrapper and the TLM values as inputs and produces a single bit that indicates which of the two possible paths (XY/YX) is less congested. This is simply done by comparing the sums of the load values along the two optional paths. This simple circuit produces a valid route selection every ATDOR operation clock cycle. Consequently, once a new source routing table is ready (e.g. 64 bits vector for 8X8 system), it is transmitted to the respective source. The table transmission and update time is negligible since ATDOR clock in much slower than that of the NoC (Section 3).

2.3 Hardware Implementation

To estimate the size and speed of our solution, we synthesized the ATDOR mechanism components for an 8x8 system using a VIRTEX 5 FPGA. Among the components are the aggregation network, the TLM and the RCM that is composed of a route selection circuit and the routing control wrapper. We implemented a simple routing control wrapper that periodically scans all possible source-destination pairs and a route selection circuit with 5 pipeline stages. The resolution of traffic load values (LRES) was 4 bits. The implementation utilized a total of 19,257 gates composed of 13,475 gates for the RCM and the TLM and 5,782 gates for the aggregation network. Maximum allowed clock speed was 252 MHz. For example, our entire routing logic for an 8x8 system requires less hardware than a single DyAD router that utilizes 25,971 gates [4].

The wiring overhead consists of one wire for every NoC horizontal link (red thin arrows in Figure 1) and LRES lines for the vertical links of the right column (red thick arrows in Figure 1). Assuming LRES is smaller than the size of the mesh, the entire wiring overhead of the ATDOR mechanism is less than one wire per NoC link.

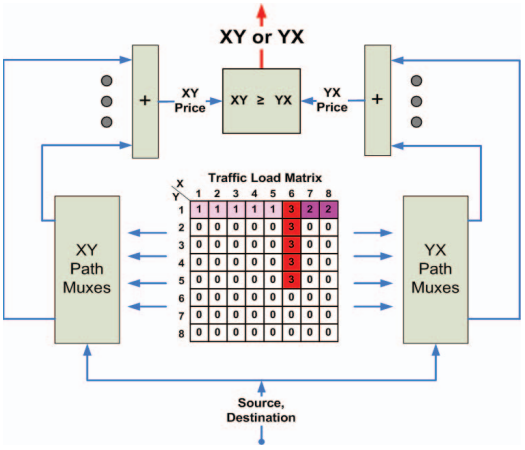


Fig. 2. Route selection circuit architecture for an 8X8 2D mesh. For every source-destination pair, the XY and YX path multiplexers select cells that belong to XY or YX paths respectively (up to 15 cells per path). Then, the sums of the load values over each path are compared and a bit is produced as a result. In the TLM above, modules (1,1) and (8,1) transmit to module (6,5) with a throughput of 1 and 2, taking the XY path. For both cases, the path selection circuit will produce "0" which indicates that YX path is less congested.

The in-router hardware implementation of the local load calculation is beyond the scope of this paper and assumed to impose ~10% of router hardware overhead as in [3].

2.4 ATDOR Scalability and Extensions

A main argument against the use of centralized routing mechanisms may be their potential scalability problems. As we demonstrate in the next section, due to its simplicity, ATDOR performs well for large designs (12X12). For future systems composed of thousands of modules we propose to employ centralized ATDOR using clustered blocks of nodes (2X2, 3X3, 4X4 etc.) instead of individual nodes. Consequently, each word in the TLM will include the aggregate load data from a respective group of routers. This way, for instance, a 32X32 mesh can be reduced to an 8X8 mesh of blocks, with 16 modules in each block.

Our implementation of ATDOR uses a simple continuous routing control mechanism, where the system periodically scans all source-destination pairs and selects the less-congested dimension ordered route for each pair. However, in systems with a relative static set of flows there is no need perform ATDOR continuously since after some initial phase, the system will converge to a load balanced routing. In order to save power, ATDOR may be stopped at that point. Finally, in systems with very few flows (e.g. Figure 2) or with few dominant flows, ATDOR may result in large fluctuations in the load as such flows are moved entirely[1] to the least congested route in each iteration. Such phenomena can be mitigated if hysteresis is added to the routing control scheme. With hysteresis, the route is changed only if the alternative route is better than the current route by a pre-defined or an adaptive threshold. Clearly, the above improvements require additional hardware resources and further research.

[1] Flows are not split to preserve in order packet delivery

3 Simulation Results

In this section we present the evaluation of ATDOR for various system sizes, traffic patterns and loads. The network and the routing mechanisms were simulated at the flow-level using a custom simulator written in C language. The time step of the simulation is a single clock cycle of the ATDOR mechanism. In our implementation, the update of the TLM with the current load occurs simultaneously with the computation of a single source routing table. This is because a single source-destination pair is computed in each clock cycle and the number of destinations is equal to the number of modules in the network. For simplicity, we assume that the re-configuration of routing tables itself is immediate since it occurs through the low latency bus [5] or the NoC itself, both operating with much faster clock than the clock of ATDOR. Unless stated otherwise, the ATDOR clock cycle is 10ns (F_{ATDOR}=100MHz). This cycle duration is easy to achieve with 65nm technology and is long enough so that the routing matrix reconfiguration affects the traffic load before the subsequent reconfiguration takes place (in a 1GHz 8X8 network there are about 640 NoC clock cycles between source routing table reconfigurations).

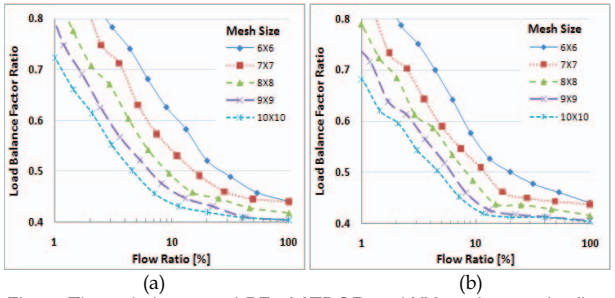
First, we explore systems which may benefit from ATDOR. Then, the performance of ATDOR is compared to distributed adaptive routing that takes its routing decisions according to local congestion, to the O1TURN oblivious routing scheme [6], and to a simple oblivious XY routing. To describe the level of load balancing, we define the Load Balancing Factor (LBF) as the normalized standard deviation (STD) of the traffic loads according to Eq. 1, where $I_{i,j}$ is the load of router i,j, and N is the number of routers in the mesh. LBF reaches a minimum of 0 for a completely balanced traffic.

$$LBF = STD\left(I_{i,j}\right) \Big/ \sum_{i=1}^{\sqrt{N}} \sum_{j=1}^{\sqrt{N}} I_{i,j} \qquad (1)$$

3.1 Basic Characteristics

In this subsection we address two related questions: (1) for which traffic matrices does ATDOR converge to a better LBF with respect to oblivious XY routing; and (2) how fast does it converge. To this end, we focus on static systems, e.g., systems with a fixed set of flows. The relative traffic intensity of each flow is uniformly chosen between 0 and 1. We term as the *flow ratio* the percentage of flows out of all possible source destination pairs. Figure 3 presents the ratio between the LBF value at steady state for ATDOR and oblivious XY routing with varying flow ratio. Figure 3a presents the LBF for a uniformly distributed traffic where in Figure 3b half of the traffic is directed towards two randomly selected "hot modules" and the rest is uniformly distributed. Each point in the graph reflects the average of 100 simulation runs. Not surprisingly, the advantage of ATDOR increases with the traffic load and the system size. We observe that the LBF of ATDOR is significantly lower than that of oblivious XY routing for a wide range of loads in moderate and large meshes, showing its superior load balancing capability.

As for the second question, we found that for all simulated systems with flow ratio higher than 1%, LBF ap-



(a) (b)

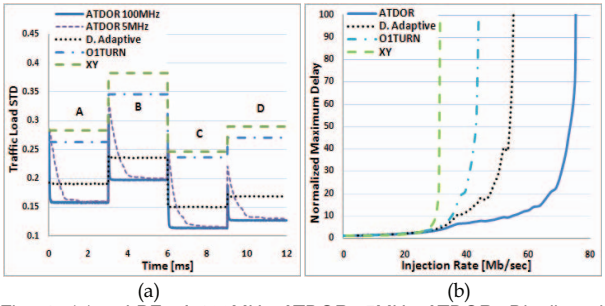
Fig. 3. The ratio between LBF of ATDOR and XY routing vs. the flow ratio for 6X6-10X10 mesh systems. (a) – Uniformly distributed traffic. (b) – Non-uniform traffic with two hot modules.

proaches a value that is within 10% of its steady state after ATDOR scans all source-destination pairs twice.

3.2 Multimode Systems

We refer to systems that switch between several modes of operation (a.k.a. "use cases") as multimode systems. Every mode is defined by a fixed set of flows. This subsection compares the performance of ATDOR in terms of delay and load balancing with that of the O1TURN scheme, oblivious XY routing and adaptive routing based on local congestion, in multimode systems for various conditions and traffic loads. ATDOR is tested with two clock speeds, 100 MHz and 5 MHz. Due to space limitation we only present results for 8X8 mesh systems. We have performed similar simulations for meshes from 5X5 up to 12X12 and observed similar behaviors.

Figure 4a presents the LBF vs. time for a system which spends a 3ms time period in each of the 4 modes of operation. Such a period is typical for a real-time full duplex video conference that requires a parallel processing of multiple voice and video encoders and decoders. The modes differ from each other by the number and intensity of flows and their spatial distribution. We test ATDOR behavior under two traffic patterns: uniform and non-uniform with two hot modules where 50% of the flows are directed to two specific modules. We present the behavior of the average LBF along time for 1000 simulation



(a) (b)

Fig. 4. (a) – LBF of 100MHz ATDOR, 5MHz ATDOR, Distributed Adaptive, and oblivious O1TURN [6] and XY vs. time for an 8X8 mesh with 4 periodic modes. The modes are defined by two parameters: (1) The fraction of active source-destination flows out of all source-destination pairs (2) The spatial distribution of traffic. Mode A – Uniformly Distributed (UD) traffic with flow ratio of 10% of . Mode B – flow ratio of 10% and 2 hot modules. Mode C – flow ratio of 20%, UD. Mode D – flow ratio of 20%, 2 hot modules. The intensity of a flow is uniformly distributed between 0 and 1. (b) – Normalized maximum delay vs. node injection rate for 2 hot modules.

runs. The flows and hot modules were reselected randomly in each simulation run for every mode. Figure 4a shows that the ATDOR improvement in load balancing over XY is from 20% to 40% better than that of distributed adaptive routing. Improvements were similar for uniform and hot-modules traffic patterns. Moreover, it shows that ATDOR control operation is effective even for F_{ATDOR} of 5MHz, several orders of magnitude slower than the clock frequency of the NoC. Figure 4b presents normalized maximum delay vs. average node injection rate for a two hot modules traffic pattern (an average of 1000 runs). The delay is modeled as an M/M/1 queue delay over the most loaded link. The vertical asymptotes in the delay curves are the saturation points of the offered load (maximal throughput). Clearly, ATDOR outperforms the other routing schemes.

3.3 Continuously Varying Systems

A system is defined as continuously varying if flows are continuously turned on and off during its operation. Similarly to the multi-mode case, we tested the performance of ATDOR for various traffic patterns and flow arrival and termination rates and observed its superiority for a wide range of traffic conditions. Our experiments are not presented due to space limitation.

4 Summary

Centralized adaptive routing has the potential to outperform distributed routing schemes based on local congestion, since a complete traffic map can be easily collected in the intra-chip environment. In this paper we introduced a centralized adaptive routing scheme for NoCs. We proposed ATDOR, a simple centralized routing system for 2D mesh, which adaptively toggles between XY and YX dimension-ordered routes for each source-destination pair. We investigated ATDOR performance using dynamic load simulations and observed that ATDOR outperforms distributed adaptive routing scheme in terms of load balancing and average latency for a wide range of traffic loads and distributions.

References

[1] E. Bolotin, Z. Guz, I. Cidon, R. Ginosar, and A. Kolodny, "The Power of Priority: NoC Based Distributed Cache Coherency", NOCS'07, 2007.

[2] R. Gindin, I. Cidon, and I. Keidar, "NoC-Based FPGA: Architecture and Routing", NOCS'07, 2007.

[3] P. Gratz, B. Grot, and S.W. Keckler, "Regional Congestion Awareness for Load Balance in Networks-on-Chip", the International Symposium on High-Performance Computer Architecture, 2008.

[4] J. Hu and R. Marculescu, "DyAD: Smart Routing for Networks on Chip", Design Automation Conference, 2004.

[5] R. Manevich, I. Walter, I. Cidon, and A. Kolodny, "Best of Both Worlds: A Bus-Enhanced NoC (BENoC)", NOCS'09, 2009.

[6] D. Seo, A. Ali, W.T. Lim, N. Rafique, and M.Thottethodi, " Near-optimal worst-case throughput routing for two-dimensional mesh networks" ,the 32nd Annual International Symposium on Computer Architecture, 2005.

[7] A. Singh, W.J. Dally, B. Towles, and A.K. Gupta, "Globally Adaptive Load-Balanced Routing on Tori", IEEE Computer Architecture Letters, vol. 3, no.1, 2004.