2011 14th Euromicro Conference on Digital System Design

A Cost Effective Centralized Adaptive Routing for Networks-on-Chip

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

Shmuel Wimer
School of Engineering
Bar-Ilan University
Ramat Gan, Israel
wimers@macs.biu.ac.il

***Abstract*— As the number of applications and programmable units in CMPs and MPSoCs increases, the Network-on-Chip (NoC) encounters unpredictable, heterogeneous and time dependent traffic loads. This motivates the introduction of adaptive routing mechanisms that balance the NoC's loads and achieve higher throughput compared with traditional oblivious routing schemes. An effective adaptive routing scheme should be based on a global view of the network state. However, most current adaptive routing schemes, following off-chip networks, are based on distributed reactions to local congestion. In this paper we leverage the unique on-chip capabilities and introduce a novel paradigm of NoC centralized adaptive routing. Our scheme continuously monitors the global traffic load in the network and modifies the routing of packets to improve load balancing accordingly. We present a specific design for the case of mesh topology, where XY or YX routes are adaptively selected for each source-destination pair. We show that while our implementation is lightweight and scalable in hardware costs, it outperforms oblivious and distributed adaptive routing schemes in terms of load balancing and average packet delay.** †

Keywords- Networks-on-Chip; adaptive routing

I. INTRODUCTION

Network-on-Chip (NoC) routing policy is among the most important considerations in on-chip network 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 **D**imension **O**rdered **R**outing (**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 the lack of a load balancing mechanism which results in a low NoC's throughput under changing or unpredictable traffic conditions. In adaptive routing, paths are dynamically adjusted to the traffic patterns and balance the utilization of links. Although adaptive routing adds considerable design complications, it is widely addressed in the NoC literature due to its potential performance benefits [14-25].

Preferably, adaptive routing should take into account the global traffic load of the network. This assertion is widely acceptable in the NoC adaptive routing literature (e.g. [19], [21-22]). However, most of these works introduce designs where routers make decisions based on local [19, 21] or regional [22] loads using local or neighbors' information. It is assumed that accounting for the full view of the network is too expensive in hardware resources and power consumption. As a result of such local routing decisions, routers might direct traffic towards remote congested areas.

Naturally, the NoC research leverages rationale, paradigms and technical solutions from the more developed off-chip networks. Along with the relevant knowledge and experience, the NoC research has sometimes adopted few conceptual limitations that are well justified in off-chip networks but may not needed in the on-chip environment. One such perception is the common notion of high complexity and cost associated with centralized control mechanisms. Global networks are characterized by long distances between the network elements and the network end-points, and dynamic network topology. Therefore, it is not practical to implement out-of-band centralized control mechanisms that continuously monitor the global state of the network and respond fast enough when control operations are needed. Consequently, off-chip networks rely almost exclusively on distributed control mechanisms that utilize in-band control information [6-7], [15-16].

NoCs differ from off-chip networks first and foremost in their physical dimensions. The typical maximal NoC system dimensions are an area of a few square centimeters and links length of few millimeters. Moreover, the NoC topology is static. These features facilitate the implementation of centralized control mechanisms that are aware of the "full network state" and are able to make optimal decisions.

In this paper we present the architecture of a novel centralized adaptive routing mechanism that takes routing decisions based on the global state of the traffic in the NoC. Our architecture is comprised of an out-of-band congestion aggregation logic and a central routing controller. As a first realization for a 2D-mesh NoC, we introduce **A**daptive **T**oggle **D**imension **O**rder **R**outing (**ATDOR**). In ATDOR, for every source-destination pair, paths are adaptively switched between XY and YX. The idea to split the traffic among XY and YX routes was first introduced in random oblivious routing algorithms [10] as O1TURN and in [11] as TXY. We extend this idea by using a central control mechanism to take state aware adaptive routing decisions.

We evaluate the hardware cost of ATDOR and compare its performance with both oblivious and distributed adaptive routings. We show that ATDOR outperforms these schemes

† A preliminary concise version of this work was published in IEEE Computer Architectures Letters, Volume 9, Issue 2, 2010.

978-0-7695-4494-6/11 $26.00 © 2011 IEEE
DOI 10.1109/DSD.2011.10

in terms of load balancing and average packet latency for a wide range of traffic patterns and NoC sizes. Moreover, we show that ATDOR requires a very small amount of hardware resources and scales well with the system size.

The paper is organized as follows: Related work is presented in section 2. Section 3 presents the concepts and architecture. Hardware implementation and area costs are discussed in section 4. Simulation results are presented in section 5. Section 6 concludes the paper. Appendix A formulates the optimal splitting of traffic between XY and YX routes that is used as an ideal benchmark.

II. Related Work

Deterministic routing schemes such as Dimension Ordered Routing (DOR, e.g. XY routing) are the simplest and the most lightweight in terms of area and therefore are very popular in the early academic NoCs (e.g. Xpipes [1], QNoC [2], Hermes [3], and AEthereal [4]) and in the industry (e.g. Tilera's Tile family [5]). For a slight cost increase, oblivious algorithms may dynamically change routing paths to improve load balancing but do not utilize information regarding the state of the network. These include Randomized Routing [8], ROMM [9] and O1TURN/TXY [10-11]. Adaptive routing schemes, despite their considerable design complications, are widely explored due to their potential to enhance performance and reliability in highly dynamic CMP and MPSoC systems. Most of the proposed adaptive routing schemes utilize distributed reaction to local traffic information, i.e. congestion information from adjacent network components [14-20]. We term these schemes as **D**istributed **A**daptive (**DA**).

Recently, several distributed routing schemes that rely on regional or global congestion information were proposed [22-24]. In **R**egional **C**ongestion **A**wareness (RCA) [22], congestion at each output port is weighted by distance, and a vicinity of 2-3 nodes is taken into account. Although RCA improves previous DA schemes, its decisions are still distributed and local, and may be wrong globally. Moreover, the advantage of RCA over simple DA routing diminishes with the system growth. In [23] each port in each router maintains delay estimations to every other node in the network. The router distributes the traffic to each destination among its output ports according to these estimations. This scheme employs a sophisticated distributed delay estimation process implemented on a separate network. Moreover, since only the distribution among admissible output ports is adaptively adjusted, packets still may take congested paths.

In [24], congestion information along source-destination paths is stored in the head packets. An alarm packet is sent back from the destination to the source if the corresponding head packet encounters congested links, and paths are changed accordingly. The scheme does not scale well since it requires hop-by-hop source routing tables at the network interfaces. There are additional drawbacks in this mechanism. First, congestion is detected only when it is actually encountered. Second, the alarm packets can also encounter congestion. Finally, the new path can also be congested. Both [23] and [24] do not present hardware implementations and area overheads so their exact scalabilities remain in question.

To the best of our knowledge, the work of Yazdi et. al. [25] is the only other work that suggests a centralized adaptive routing mechanism based on a global traffic load map. In [25] the Floyd-Warshal algorithm is used to compute the shortest path routing permutation using the traffic load-map. The main drawbacks of this paper are the excessively large time complexity of the routing computation procedure ($\sim O(n^3)$) and the lack of hardware implementation and evaluation of the area costs. Moreover, there is a concern regarding the correctness of the routing calculation. Floyd-Warshal algorithm is applicable for directed graphs with constant weights and is used on a graph where the weights that represent traffic are not constant.

ATDOR combines the advantages learned from previous works. First, we use the simple DOR routing. Second, we propose a centralized architecture that makes globally optimal routing decisions and with a $\sim O(n^2)$ computation complexity for all source-destination pairs in the system. Third, our solution has considerable smaller area compared with the previously proposed global adaptive routing. Finally, the paper is first to introduce a complete centralized routing solution implementation in RTL and to present accurate area costs.

III. Concepts and Architecture

We concentrate our discussion on 2D mesh topology. The centralized adaptive NoC routing mechanism is composed of a feedback module and a control module. The feedback module monitors the traffic load across the network links and aggregates it into four separate central traffic load maps for links directed to the east, west, south and north directions at each router. The control module adjusts the routing according to the most updated traffic load maps, in order to maximize load balancing and to reduce congestion. The architecture of the scheme for a 4X4 2D mesh is presented in figure 1.

A. Feedback

The feedback module is composed of three functional layers. The first layer is implemented at the routers and is responsible for monitoring the load of the network links. The traffic load of the downstream links is measured in each router. K-bit traffic load samples are stored in four registers, one for each link. There are several common metrics which may be 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 [22]. The second layer is responsible for aggregating the local load measurements into the centralized traffic load maps. 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). There are four separate aggregation networks for E, W, S and N links. The throughput of this

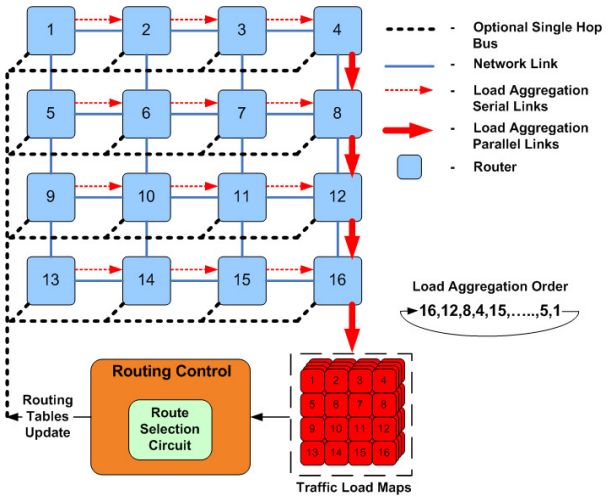


Figure 1. Architecture of a 4x4 2D mesh.

implementation is 4K bits per clock cycle. Assuming that the traffic load measurement resolution K is lower than the number of columns in the 2D mesh, the wiring overhead of the aggregation networks is at most 4 additional wires per NoC link. The third layer is the **T**raffic **L**oad **M**aps (**TLM**s). There are 4 TLMs that includes local load measurements for E, W, S and, N outgoing links of each router reflecting the most updated global traffic congestion information. The TLMs utilize double buffering: while the local load measurements are collected to the first TLM buffer, the routing calculation is conducted using the values previously stored in the second buffer. Hence, the load aggregation process works in iterations where in each iteration all K bit values from all links are brought to the first buffer and copied within one clock cycle to the second buffer. Figure 2 drills into the aggregation network.

B. Routing Control Architecture

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 **R**outing **C**ontrol **M**odule (**RCM**) is responsible for calculating the routing tables according to the content of the TLMs. The routing between source destination pairs is dynamically adjusted to improve load balancing and reduce congestion. Following an update of the TLMs, the RCM examines a set of source-destination pairs and re-computes the least congested route for each pair. Once the RCM completes the route computation for all the destinations of a single source, it updates the corresponding source routing table. The sequence of flows that are re-routed, the timing between routing adjustments and the update rate of the TLMs are determined by the control unit. Conceptually, the control unit can periodically scan all the source-destination pairs or implement dynamic prioritization among the active flows. The update of the routing tables is performed through the NoC itself. In systems with highly fluctuating traffic, it is possible to prioritize the control

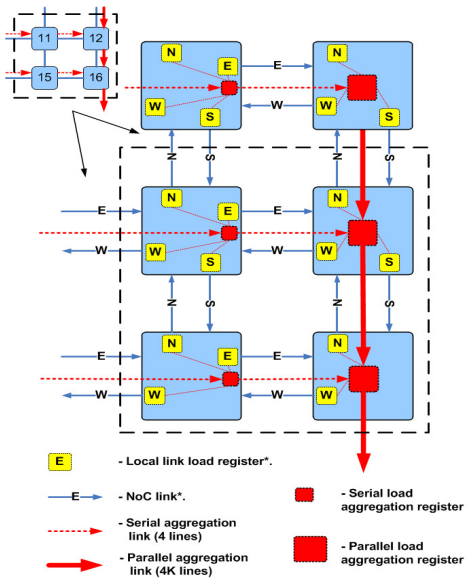


Figure 2. Zoom into the traffic load aggregation network. E, W, S and N are the directions East, West, South and North.

traffic [12] or to carry it over an out of band low latency bus [13]. This will to accomplish a faster feedback loop.

Finding in real time the least congested route is a complex task even if we confine ourselves to shortest path routes. Moreover, the use of complex routes requires the source routing tables to include detailed hop-by-hop path information and to add an expensive deadlock avoidance mechanism. In order to simplify the solution and reduce hardware costs, we define a simpler adaptive 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. The most congested links of each route are compared, and the route with less loaded "most congested link" is preferred. This allows using one bit source routing tables at the sources. Moreover, only two VCs are needed (one for the XY routes and the other for YX routes) to avoid deadlocks. Finally, it allows the implementation of a fast and simple route selection circuit at the RCM. In our implementation RCM is only familiar with the measured global traffic congestion status as it is stored in the TLMs. It is not aware of the source-destination flows that form this status. Therefore, all possible source-destination pairs are treated as having an equal rate.

The basic routing control unit scans periodically and continuously all possible source-destination pairs throughout the whole system operation period. This basic policy may converge to a continuous route change oscillations around the optimal working point even if the source to destination flows are static. Two improvements are added to the basic technique. First, we allow a route change only if the current route is worse than the alternative route multiplied by a routing threshold factor $0<\alpha\leq 1$ (eq. 1). We elaborate regarding the effects of α on the performance of ATDOR in section 5.

$$if\,(Current\ Route = XY)$$
$$NextRoute = \begin{cases} YX & if\ MAX[Load_{YX}] \le \alpha MAX[Load_{XY}] \\ XY & if\ MAX[Load_{YX}] > \alpha MAX[Load_{XY}] \end{cases}$$
$$elseif\,(Current\ Route = YX) \qquad (1)$$
$$NextRoute = \begin{cases} XY & if\ MAX[Load_{XY}] \le \alpha MAX[Load_{YX}] \\ YX & if\ MAX[Load_{XY}] > \alpha MAX[Load_{YX}] \end{cases}$$

Second, we add a re-routings counter for each source-destination pair. We use the notation $C_{I,J}$ for a counter that counts routing modifications of the path $P_{I,J}$ between source I and destination J. $P_{I,J}$ can be either XY or YX. When $C_{I,J}$ reaches a pre-defined limit $L_{I,J}$, the routing of the corresponding flow I,J cannot be changed any longer. The process converges gradually and terminates when all $C_{I,J}$ reach their limits or there were no re-routings throughout a single pass over all the couples (I,J) (i.e. "ATDOR control iteration"). Our experiments indicate that the mechanism performs very well for $C_{I,J}$ of 3 bits ($L_{I,J} \le 7$). $L_{I,J}$ set to vary from 1 to 7 (eq. 2):

$$L_{I,J} = 1 + \left[(I + J) \bmod 7 \right] \qquad (2)$$

The two proposed improvements of the basic ATDOR mechanism help to prevent unnecessary fluctuations in routing. Limiting of re-routing improved the energy consumption. The hardware overhead of the improvements, assuming $C_{I,J}$'s of 3 bits, is approximately 3 bits per source-destination pair. Although it grows quadratically with the number of modules, it does not exceed several 10's of Kbits for reasonable size systems.

We term the time required for a single ATDOR control iteration as $T_{CONTROL}$. A new control iteration can be initiated by resetting the counters. We examine three counters reset policies. In the first, termed "Oblivious Periodic Control",

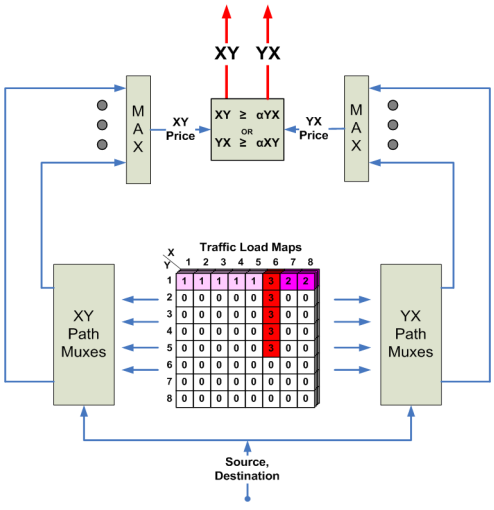


Figure 3. Route selection circuit architecture for an 8X8 2D mesh. For every source-destination pair, the XY and YX path multiplexers select cells that belong to XY or YX paths respectively (up to 15 cells per path). Then, the maximum load values over each path are compared and 2 bits are produced as 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. Assuming α=0.75, for both cases, the path selection circuit will produce "1,0" which indicates that YX path is less congested and should be preferred.

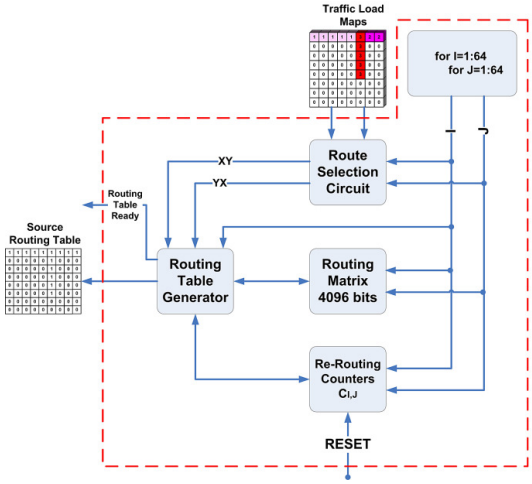


Figure 4. Routing control module for an 8X8 2D mesh system.

counters are reset periodically every $T_{RESET} >> T_{CONTROL}$. In the second, called "Manual Control", the reset occurs following an external interrupt, that can be initiated by the application when the operation mode changes. In the third reset policy termed "Adaptive Control", the RCM counts the paths that would be re-routed if the associated counters would not reach their limits, and resets all counters if the number of such paths exceeds a deterministic or dynamically calculated threshold.

The architecture of a combinational pipelined route selection circuit is presented in figure 3. The proposed circuit receives the coordinates (I,J) of source and destination modules from the routing control wrapper and the TLM values as inputs and produces two bits that indicate whether the alternate XY or YX path is less congested by the factor of eq. 1. This is simply done by comparing the maximum load values along the two optional paths. This simple circuit produces a best route selection every ATDOR operation clock cycle. The decision whether to re-route a given flow is made by comparing the output of the routing circuit and the routing matrix that contains the current routes. The architecture of the whole RCM for an 8x8 2D mesh is presented in figure 4.

C. *ATDOR Scalability*

The intuitive argument against the use of centralized routing mechanisms is limited scalability. As we report in section 5, due to its simplicity, ATDOR performs well for practical large chip designs (12X12). For future systems composed of thousands of modules we still propose to employ centralized ATDOR using load information measured at clustered blocks of nodes (2X2, 3X3, 4X4 etc.) instead of individual nodes. Consequently, each word in the TLM will include the aggregate load from a respective set of routers. Hence, for instance, a 32X32 mesh can be reduced to an 8X8 mesh of blocks, with 16 modules in each block.

IV. HARDWARE IMPLEMENTATION

To approximate the typical size and speed of our solution, we synthesized the ATDOR mechanism

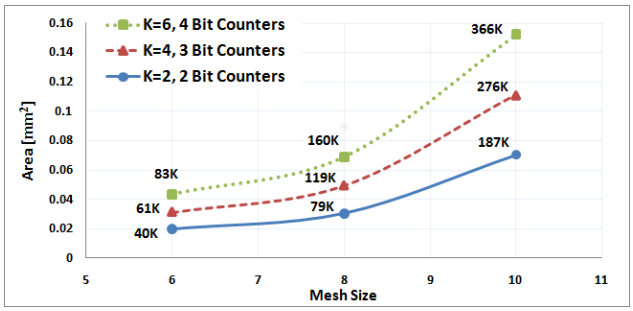


Figure 5. ADTOR equivalent gates count and estimated area at the 45nm technology node vs. the size (i.e. number of columns) of the 2D mesh. The numbers next to the markers are the equivalent gate counts in 1000's.

components for 6x6, 8x8 and 10x10 systems using the Xilinx xc5vlx50t 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 implement the routing control module as it was described in sub-section 3.b (figure 4) with a "Manual Control" counters reset policy and a route selection circuit with 5 pipeline stages. In order to evaluate the contribution of the local load representation resolution (K) and the size of counters $C_{I,J}$ to the total area, ATDOR is implemented for three different pairs of values of these parameters. In figure 5 we present the total equivalent gate count (a metric that is presented by the XST synthesis tool of Xilinx) and the estimated area for 45nm technology node of the entire ATDOR mechanism. The area is estimated using 45nm technology node SRAM and logic cell area values from the 2007 annual report of ITRS [31]. $C_{I,J}$ assumed to utilize SRAM cells while the rest assumed to be regular logic cells. The largest implementation among all combinations (i.e. 10x10 system, K=6, 4 bit $C_{I,J}$) utilized an area of ~0.15mm^2, ~0.05% of the area of a typical 300mm^2 die. Maximum allowed clock speeds were between 117MHz-155MHz for all implementations (for VIRTEX 5, 65nm FPGA).

The wiring overhead consists of four wires for every NoC horizontal link (thin dashed arrows in figure 1) and 4K lines for the vertical links of the right column (thick arrows in figure 1). Assuming K is smaller than the size of the mesh, the entire wiring overhead of the ATDOR mechanism is less than four wires per NoC link. 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 [22].

V. SIMULATIONS

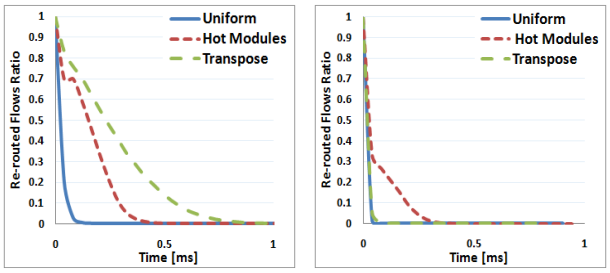
In this section we evaluate the performance of ATDOR versus previously introduced routing schemes for synthetic traffic patterns and SPLASH-2 [29] and PARSEC [30] benchmarks running on **D**ynamic **N**on-**U**niform **C**ache **A**rray (**DNUCA**) **C**hip **M**ulti-**P**rocessor (**CMP**). We present simulation results for an 8X8 2D mesh. Similar results were observed for 5X5-12X12 systems. 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 the routing tables is immediate since it is applied through a low latency bus [13] or the NoC itself, both operating with a much faster clock than that of ATDOR. The ATDOR clock cycle is 10ns (F_{ATDOR}=100MHz). This cycle is easy to achieve even 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). To illustrate the influence of the parameter α from eq. 1, two values were used (15/16, 3/4). $L_{I,J}$ follow eq. 2.

We compare the performance of ATDOR for both values of α with the following alternatives: RCA Quadrant routing– the high performance form of the Regional Congestion Awareness routing scheme [22], distributed adaptive (DA) routing that makes its routing decisions according to local congestion [20], O1TURN oblivious routing scheme [10], deterministic XY routing and an optimal DOR solution obtained by solving the optimization problem that is presented in Appendix A. The NoC is assumed to run with a clock of 1GHz. We model links as M/M/1 queues with a minimum latency of 1ns (1 cycle per hop) and calculate average packet delay. Three synthetic traffic patterns were used: uniform traffic pattern; transpose traffic pattern where modules transmit only to modules found in the diagonal quadrants; and a hot spot pattern with 4 hot modules receiving and transmitting ~25 times more data than regular ones. We assume equal capacity links. The presented results of the synthetic traffic patterns are averaged over 1000 simulation runs.

A. *Control Iteration Duration*

We define control iteration as the period from a reset of counters $C_{I,J}$ till all $C_{I,J}$ reach their limits $L_{I,J}$ or until no-routing modification is performed throughout a single pass of the RCM over all the flows $F_{I,J}$. We use the notation $IF_{I,J}$ for the intensity (i.e. session bandwidth in Mbits/sec) of $F_{I,J}$. In this sub-section, the system is single-mode quasi-static (i.e. $IF_{I,J}(t)$=const). The number of flows that are re-routed at every pass (a.k.a. **CSI** – **C**ontrol **S**ub-**I**teration) over all I,J's is measured. Figure 6 presents this number normalized with its initial value vs. time for uniform, hot-module and transpose traffic patterns. For an 8X8 system, each CSI takes 64X64 ATDOR clock cycles, and the time scale correlates with ATDOR clock of 100 MHz.

We refer to systems that switch between several modes of operation as multimode systems. Every mode is defined by a fixed set of flows. Such systems, in general, are expected to spend at least several milliseconds in each mode. In figure 6 we observe that for the worst case of α=15/16 the ATDOR control iteration demands less that 1ms for the worst test-case (transpose traffic) and less than 0.5 ms for the



(a) – α=15/16 (b) – α=3/4

Figure 6. Control iteration duration – the number of re-routed flows every CSI normalized to the number of flows that were re-routed at the first CSI vs. time for an 8X8 2D mesh and ATDOR clock of 100 MHz.

more realistic hot-module patterns. For α=3/4 and α=15/16 ATDOR terminates when most loaded links of all XY and YX pairs are within a range of 25% and 6.25% from each other, respectively. Therefore, as expected, the convergence is much faster for α=3/4 since a proximity of 25% is much faster achieved.

A control period of 0.5ms is expected to be sufficient for CMPs running real-time multimedia, multimode applications. However, for large systems (i.e.>12X12), this period might exceed the cost-effective point for such systems. In that case, a utilization of the hierarchical ATDOR architecture (section 2.3) should be considered.

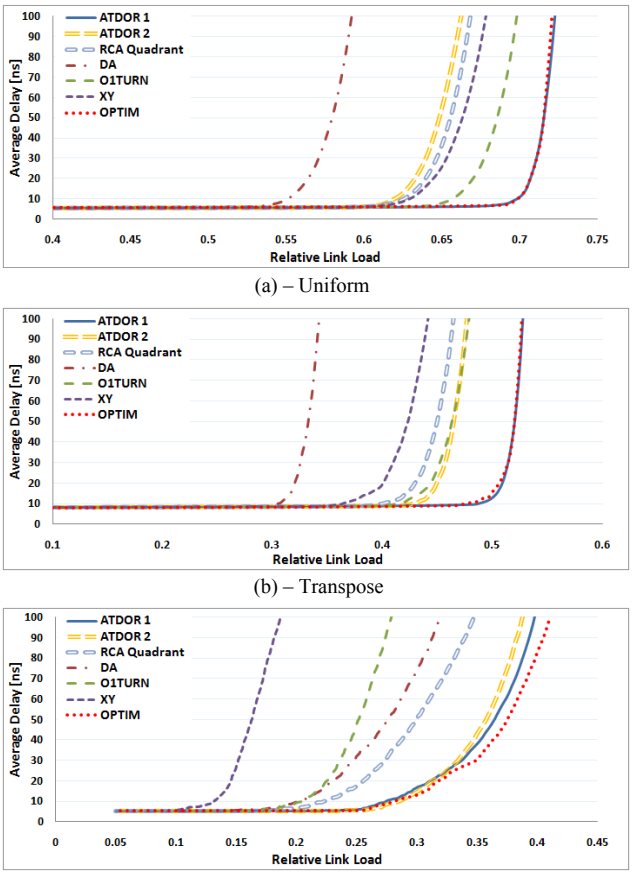
B. Average Delay – Synthetic Traffic Patterns

In this sub-section we compare the performance of ATDOR in terms of average packet delay with that of the RCA Quadrant [22], DA [20], O1TURN [10] and, XY routing schemes, and the "ideal" solution from Appendix A (termed OPTIM), for the traffic patterns described before. For each simulation run, we sample the TLMs of all routing schemes after the completion of the ATDOR control iteration. Then, we calculate the average packet delay for each set of TLMs for various link capacities. The results are presented in figure 7. Average packet delay is plotted vs. Relative Links Load (RLL) that is defined in eq. 7:

$$RLL = \frac{Average\ Link\ Load}{Link\ Capacity} = \frac{\sum_{D}\sum_{x,y} TLM_{x,y,D}}{Link\ Capacity} \qquad (3)$$

$D = East, West, North, South$

As seen in figure 7, ATDOR 1 (α=15/16) considerably outperforms the other schemes for all the use-cases. Moreover, it introduces a very close average delay to the OPTIM "ideal" XY/YX confined solution. ATDOR 2 (α=3/4) performs similarly to ATDOR 1 for the hot- modules use- case, and falls behind for the uniform and the transpose traffic patterns. The latter finding illustrates the important impact of α. For a given $F_{I,J}$, ATDOR 2 does not switch routing if the traffic load values of most loaded links of XY and YX paths are within a range of 25% from each other. In uniform and transpose traffic patterns with many sessions with similar $IF_{I,J}$ there is a statistical symmetry between XY and YX options. Moreover, the contribution of each $IF_{I,J}$ to the accumulated load along the links is small. Therefore α has to be close enough to one to allow an appropriate load



(a) – Uniform

(b) – Transpose

(c) – 4 Hot Modules

Figure 7. Average delay – synthetic patterns. ATDOR 1: α=15/16, ATDOR 2: α=3/4, DA: distributed adaptive, OPTIM: the solution of the optimization problem from Appendix A.

balancing. On the other hand, in the hot-module pattern, ATDOR 2 performed well thanks to the increased diversity in flows physical distribution and bandwidth (i.e. $IF_{I,J}$).

C. ATDOR for CMP DNUCA

Finally, we evaluate the performance of ATDOR and the other routing schemes with traffic patterns formed in a large scale CMP with 8 CPUs and a DNUCA. The DNUCA system is modeled using the Simics [28] simulator running the SPLASH-2 [29] and PARSEC [30] benchmarks. Traffic patterns (i.e. sets of $IF_{I,J}$) are obtained by mapping the resulting traces onto the 2D mesh topology that is presented in figure 8. For our 8X8 2D Mesh topology, we define **N**etwork **S**aturation **RLL** (**NSRLL**) as the RLL that yields a packet average delay of 100ns. We examined the performance of the routing schemes on 6 different applications. The respective NSRLLs are presented in figure 9. Control iteration durations for both ATDOR 1 and ATDOR 2 are presented in figure 10. Here, we observe that ATDOR 2 (α=3/4) is the best performer for all the benchmarks. The number of active flows in the benchmarks (i.e. $IF_{I,J}$>0) is much lower compared to the synthetic patterns from the previous sub-section. Therefore a higher hysteresis value has to be set between the XY and YX options to

C1	$1	$2	$3	$4	$5	$6	$7
$8	C2	$9	$10	$11	$12	$13	$14
$15	$16	C3	$17	$18	$19	$20	$21
$22	$23	$24	C4	$25	$26	$27	$28
$29	$30	$31	$32	C5	$33	$34	$35
$36	$37	$38	$39	$40	C6	$41	$42
$43	$44	$45	$46	$47	$48	C7	$49
$50	$51	$52	$53	$54	$55	$56	C8

Figure 8. 2D Mesh CMP with 8 CPUs and a DNUCA – C1-C8 are CPUs arranged in the main diagonal of the mesh, $1-$56 are cache banks.

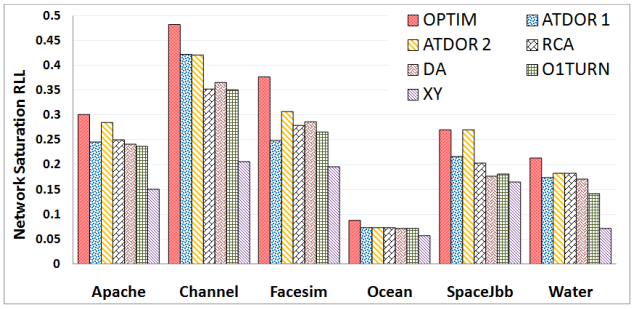
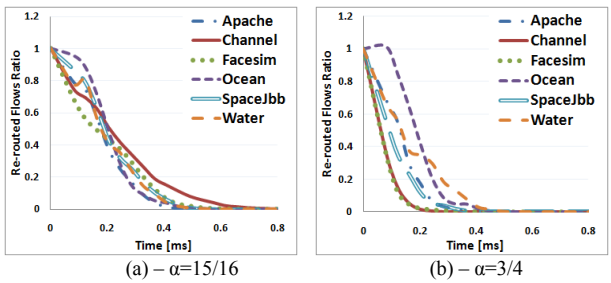


Figure 9. Network saturation relative links load for Splash 2 and Parsec benchmarks – ATDOR 1: α=15/16. ATDOR 2: α=3/4.

prevent re-routings that result in higher congestion. Control relaxation (figure 10) behaves similarly to the synthetic case. The simulations illustrate the trade-off presented by the selection of the parameter α for the performance of ATDOR. When calculating the preferred route between source I and destination J, ATDOR is not aware whether there is an active session (flow) between these modules and how intense it is since it sees only the cumulative link loads as they are reflected in the TLMs. Therefore, it may change routing such that the new path of the flow is more congested. On one hand, for lower α, less re-routings result in increased congestion. However, on the other hand, it is obvious that low α limits the ability of ATDOR to achieve load balance. Dynamic adjustment of α has to be supported to cover both uniform and highly diverse traffic patterns sharing the same system in different time divisions (TDM). The parameter α can be adaptively adjusted throughout the operation of the system, however, this discussion is beyond the scope of this paper due to space limitation.



(a) – α=15/16 (b) – α=3/4

Figure 10. Splash 2 and Parsec benchmarks - control iteration duration.

VI. Summary

Centralized adaptive routing has the potential to outperform distributed adaptive routing schemes based on local and global congestion measurements, since only centralized mechanisms are capable of making globally near-optimal decisions. Unlike off-chip networks, 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, lightweight and scalable 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 oblivious and distributed adaptive routing schemes in terms of load balancing and average latency for a wide range of synthetic and real traffic loads and distributions. By comparing with results of a near-ideal algorithm we also show that ATDOR exploits most of the benefit that can be gained using an adaptive XY/YX routing algorithm.

Appendix A – Optimal DOR Solution

We present a technique to calculate an optimal load balanced routing solution restricted to XY and YX routes (i.e. DOR) for 2D mesh. While this technique is too complex for hardware implementation, it is developed as a yard stick for comparing ATDOR to the "ideal" solution. In the next section, this technique is used to evaluate the quality of the routing solutions obtained by ATDOR. Assuming that the traffic pattern is known, we define an optimization problem in order to find the lowest achievable minimum load on the most loaded link. The problem is formally defined in this section and solved numerically in the next section for several use-cases. The following notation is used:

(I, J) – Source, Destination indexes (figure 1).

$F_{I,J}$ – A flow between source I and destination J.

$IF_{I,J}$ – The intensity of $F_{I,J}$, can be defined also as session bandwidth in Mbits/sec.

$Q_{I,J}$ – The fraction of $IF_{I,J}$ routed XY route.

$P_{I,J}$ – The fraction of $IF_{I,J}$ routed YX route.

$LINK_{X,Y,D}$ – An output link in the direction D of a router located in coordinates X,Y in the 2D mesh. D can be west, east, north or south.

$Q_{I,J}$ and $P_{I,J}$ satisfy the following set of constraints:

$$Q_{I,J} + P_{I,J} = 1; 0 \le Q_{I,J} \le 1 \quad ; 0 \le P_{I,J} \le 1 \tag{4}$$

The values of $Q_{I,J}$ and $P_{I,J}$ define the routing scheme. For XY routing, $Q_{I,J}=1$ and $P_{I,J}=0$ $\forall$ I,J. For YX routing, $Q_{I,J}=0$ and $P_{I,J}=1$. In the O1TURN scheme [5], $\forall$ I,J, $Q_{I,J}=P_{I,J}=0.5$. ATDOR produces routing patterns where $Q_{I,J}$ and $P_{I,J}$ are integers (0 or 1). We define a continuous optimization problem where $Q_{I,J}$ and $P_{I,J}$ can get any real value as long as long as the constraints of equation 4 are satisfied for two reasons. First, its solution is equal to or better than the solution of the discrete problem; second such continuous optimization problem can be solved using linear

programming in a polynomial computation time. The following 2 equations are required to define the optimization problem:

$$M_Q = M_Q(I,J,X,Y,D) = \begin{cases} 1 & Q_{I,J} \in LINK_{X,Y,D} \\ 0 & Q_{I,J} \notin LINK_{X,Y,D} \end{cases}$$
$$M_P = M_P(I,J,X,Y,D) = \begin{cases} 1 & P_{I,J} \in LINK_{X,Y,D} \\ 0 & P_{I,J} \notin LINK_{X,Y,D} \end{cases} \quad (5)$$

$$Most\ Loaded\ Link = MLL(Q_{I,J}, P_{I,J}) = \max_{X,Y,D}\left(\sum_{I,J} M_Q Q_{I,J} + \sum_{I,J} M_P P_{I,J}\right) \quad (6)$$

Finally, the problem is defined as follows: find the set $(Q_{I,J}, P_{I,J})^*$ such that:

$$\left(Q_{I,J}, P_{I,J}\right)^* = \arg\min_{(Q_{I,J}, P_{I,J})} MLL(Q_{I,J}, P_{I,J}) \quad (7)$$

To solve the optimization problem, we used the SNOPT 7.2 LP solver [27] under the AIMMS environment [26]. The solver iteration limit was set to 20,000. The $(Q_{I,J}, P_{I,J})^*$ that were obtained as a result were used to build a virtual TLM and calculate average packet delay. It important to point out that our measurement of performance (i.e. average packets delay) is not completely equivalent to the optimization variable of the problem that we defined. Thus, ATDOR or other routing schemes might yield even better performance results in terms of average packets delay than the optimal solution with regard to maximum link load. Yet, the solution of the problem can be accounted as a qualitative solution of similar enough problem obtained by an honorable amount of off-line computation.

REFERENCES

[1] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip", IEEE circuits and systems magazine, 2004.

[2] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network-on-chip", The journal of systems architecture, special issue on networks on chip, 2004.

[3] F. Moraes et al., "HERMES: an infrastructure for low area overhead packet-switching NoC", VLSI Journal, 2004.

[4] K. Goossens, J. Dielissen, and A. Radulescu, "AEthereal network on chip: concepts, architectures, and implementations", IEEE design and test of computers, 2005

[5] Anant Agarwal et al., "Tile processor: embedded multicore for networking and multimedia", Proc. of hot chips: symposium on high performance chips, 2007.

[6] G. G. Finn, "Routing and addressing problems in large metropolitan-scale intemetworks", Tech. rep. ISI/RR-87-180, information sciences institute, 1987.

[7] P. R. Bell and K. Jabbour, "Review of point-to-point network routing algorithms", IEEE communication magazine, 1986.

[8] L. G. Valiant , G. J. Brebner, "Universal schemes for parallel communication", Proc. of the thirteenth annual ACM symposium on theory of computing, 1981.

[9] T. Nesson , and S. L. Johnsson, "ROMM routing on mesh and torus networks", Proc. of the seventh annual ACM symposium on parallel algorithms and architectures, 1995.

[10] 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.

[11] R. Gindin, I. Cidon, and I. Keidar, "NoC-based FPGA: architecture and routing", NOCS'07, 2007.

[12] E. Bolotin, Z. Guz, I. Cidon, R. Ginosar, and A. Kolodny, "The power of priority: NoC based distributed cache coherency", NOCS'07, 2007.

[13] R. Manevich, I. Walter, I. Cidon, and A. Kolodny, "Best of both worlds: a bus-enhanced NoC (BENoC)", NOCS'09, 2009.

[14] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks", IEEE transactions on parallel and distributed systems,1993.

[15] W.J. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels", IEEE trans. on parallel and distributed systems, 1993.

[16] C. J. Glass and L. M. Ni, "The turn model for adaptive routing", Journal of ACM, 1994.

[17] G. M. Chiu, "The odd-even turn model for adaptive routing", IEEE trans. on parallel and distributed systems, 2000.

[18] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles, "GOAL: a load balanced adaptive routing algorithm for torus networks", The 30th annual international symposium on computer architecture ISCA'03, 2003.

[19] J.Hu and R. Marculescu, "DyAD: smart routing for networks on chip", Design automation conference, 2004.

[20] M. Li, Q. A. Zeng, and W. B. Jone, "DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip", DAC '06: Proc. of the 43rd annual conference on design automation, 2006.

[21] A. Singh, W.J. Dally, B. Towles, and A.K. Gupta, "Globally adaptive load-balanced routing on tori", IEEE computer architecture letters, 2004.

[22] 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.

[23] R. S. Ramaujam, and B. Lin, "Destination-based adaptive routing on 2D mesh networks", Proc. of the 6th ACM/IEEE symposium on architectures for networking and communications systems, 2010.

[24] L.P. Tedesco, T. Rosa, F. Clermidy, N. Calazans, and F.G. Moreas," Implementation and evaluation of a congestion aware routing algorithm for networks-on-chip", SBCCI '10: Proc. of the 23rd symposium on integrated circuits and system design, 2010.

[25] M.A. Yazdi, M. Modarressi, and H. Sarabi-Azad, "A load-balanced routing scheme for noc-based systems-on-chip.", DMEMS '10: Proc. of the 2010 first workshop on hardware and software implementation and control of distributed MEMS, 2010.

[26] AIMMS optimization software for mathematical programming – http://www.aimms.com.

[27] SNOPT 7.2 solver - http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm

[28] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, and G. Hallberg, "Simics: a full system simulation platform", IEEE computer, 2002.

[29] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta., "The SPLASH-2 programs: characterization and methodological considerations", In proceedings of the 22nd annual international symposium on computer architecture, 1995.

[30] C. Bienia, S. Kumar, J. P. Singh and K. Li, "The PARSEC benchmark suite: characterization and architectural implications", Proc. of the 17th international conference on parallel architectures and compilation techniques, 2008.

[31] ITRS – International technology roadmap for semiconductors, http://www.itrs.net