A Multi-Path Routing Strategy with Guaranteed In-Order Packet Delivery and Fault-Tolerance for Networks on Chip

Srinivasan Murali[⋆], David Atienza[§†], Luca Benini[‡], Giovanni De Micheli[†]
[⋆]CSL/Stanford University, Stanford, USA, smurali@stanford.edu
[§]DACYA/UCM, Madrid, Spain, datienza@dacya.ucm.es
[†] LSI/EPFL, Lausanne, Switzerland, {david.atienza, giovanni.demicheli}@epfl.ch
[‡]DEIS/University of Bologna, Bologna, Italy, lbenini@deis.unibo.it

ABSTRACT

In this work we present a multi-path routing strategy that guarantees in-order packet delivery for *Networks on Chips (NoCs)*. We present a design methodology that uses the routing strategy to optimally spread the traffic in the NoC to minimize the network bandwidth needs and power consumption. We also integrate support for tolerance against transient and permanent failures in the NoC links in the methodology by utilizing spatial and temporal redundancy for transporting packets. Our experimental studies show large reduction in network bandwidth requirements (36.86% on average) and power consumption (30.51% on average) compared to single-path systems. The area overhead of the proposed scheme is small (a modest 5% increase in network area). Hence, it is practical to be used in the on-chip domain.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems; C.4 [**Performance of Systems**]: Design studies
General Terms: Design, Measurement, Performance.
keywords: Systems on Chip, networks on chip, routing, multi-path, fault-tolerance, re-order buffers, flow control.

1. INTRODUCTION

Scalable *Networks on Chips (NoCs)* are needed to provide high bandwidth communication infrastructure for SoCs [1]-[3]. The use of NoCs facilitate applying network error resiliency techniques to tolerate transient and permanent errors in interconnects.

For routing packets in the NoC, either a single-path can be used for all the packets from a source to a destination or multiple paths can be utilized. When compared to single-path routing, the multi-path routing scheme improves path diversity, thereby minimizing network congestion and traffic bottlenecks. Reducing the traffic bottlenecks leads to lower required NoC operating frequency, as traffic is spread evenly in the network. A reduced operating frequency translates to a lower power consumption in the NoC. Another important property of the multi-path routing strategy is its spatial redundancy for transporting a packet in the on-chip network.

Many of today's NoC architectures are based on single-path routing. This is because, with multi-path routing, packets can reach the destination in an out-of-order fashion due to the difference in path lengths or due to difference in congestion levels on the paths. The re-order buffers needed at the receiver for ordering the packets have large area and power overhead and deterministically choosing the size of them is infeasible in practice. The re-order buffers, unless they have infinite storage capacity, can be full for a particular scenario and can no longer receive packets. This leads to dropping of packets to recover from the situation and requires end-to-end ACK/NACK protocols for resuming the transaction. However, such protocols have significant overhead in terms of network resource usage and congestion [13]. Thus, they are not commonly used in the NoC domain.

In this work we present a multi-path routing strategy with guaranteed in-order packet delivery (without packet dropping) for on-chip networks. We present a method to split the application traffic across multiple paths to obtain a network with minimum power consumption. We integrate reliability constraints in our multi-path design methods to provide a reliable NoC operation with least increase in network traffic. Experiments on several benchmarks show large power savings for the proposed scheme when compared to traditional single-path schemes and multi-path schemes with re-order buffers. The area overhead of the proposed scheme is very small (only 5% increase in network area). Hence, it is usable in the on-chip domain.

Many works on mapping of applications onto NoC architectures have considered the routing problem during the NoC design phase [5]-[8]. The adaptive routing schemes presented in [9] and [10], assume that the architectural support needed for such routing schemes (such as packet re-order buffers) are available in the NoC. Several works in the multi-processor field have focused on the design of efficient routing strategies [19]- [22]. Several research works have focused on designing reliable NoC systems [11]-[18]. In [14], fault-tolerant stochastic communication for NoCs is presented. The use of non-intersecting paths for achieving fault-tolerant routing has been utilized in many designs, such as the IBM Vulcan [19]. The use of temporal and spatial redundancy in NoCs to achieve resilience from transient failures is presented in [18].

2. ROUTING WITH IN-ORDER DELIVERY

In this section, we present the conceptual idea of the multi-path routing strategy with in-order packet delivery. For analysis purposes, we define the NoC topology and the traffic flow paths as:

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

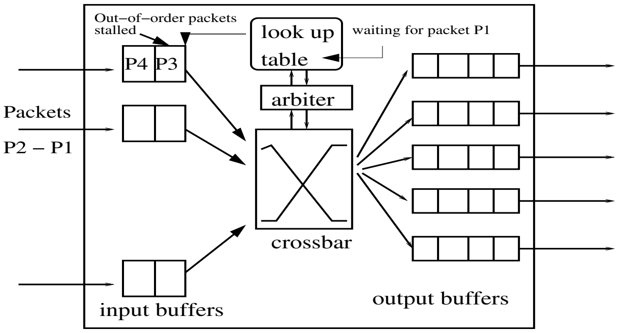


Figure 1: Switch design to support multi-path routing

DEFINITION 1. *The topology graph is a directed graph $G(V, E)$ with each vertex $v_k \in V$ representing a switch/Network Interface (NI) in the topology and the directed link (or edge) $e_l \in E$ representing a direct communication between two switches/NIs.*

We represent the traffic flow between a pair of cores in the NoC as a commodity i, with the source switch/NI of the commodity being s_i and the destination of the commodity being d_i. Let the total number of commodities be I. The rate of traffic transferred by commodity i is represented by r_i.

DEFINITION 2. *Let the set SP_i represent the set of all paths for the commodity i, $\forall i \in 1 \cdots I$. Let P_i^j be an element of SP_i, $\forall j \in 1 \cdots |SP_i|$. Thus P_i^j represents a single-path from the source to destination for commodity i. Each path P_i^j has a set of links.*

We define a set of paths to be *non-intersecting* if the paths originate from the same source vertex but do not intersect each other in the network, except at the destination vertex. Consider packets that are routed on the two *non-intersecting* paths. Note that with worm-hole flow control [19], packets of a commodity on a particular path are in-order at all time instances. However, packets on the two different paths can be out-of-order. Therefore, we need a mechanism to re-order the packets at the path *re-convergent* nodes to maintain the packet ordering.

To implement the re-ordering mechanism at network re-convergent nodes, the following architectural changes to the switches/NIs of the NoC are required (shown in Figure 1). Even though the methodology presented in this paper is general, for illustrative purposes we assume that the component architectures are based on the design presented in [4]. To support multi-path routing, individual packet identifiers are used for packets belonging to a single commodity. At the *re-convergent* switch, we use a look-up table to store the identifier of the next packet to be received for the commodity. Initially (when the NoC is reset), the identifiers in the look-up tables are set to 1 for all the commodities. When packets arrive at the input of the *re-convergent switch*, the identifier of the packet is compared with the corresponding look-up table entry. If the identifiers match, the packet is granted arbitration and the look-up table identifier value for this commodity is incremented by 1. If the identifiers do not match, it is an out-of-order packet and access to the output is not granted by the arbiter, and it remains at the input buffer.

As the packets on a particular path are in-order, the mechanism only stalls packets that would also be out-of-order if they reach the switch. Due to the disjoint property of the paths reaching the switch, the actual packet (matching the identifier on the look-up table) that needs to be received by the switch is on a different path. As a result, such a stalling mechanism (integrated with *credit-based* or *on-off* flow control mechanisms [19]) does not lead to packet dropping, which is encountered in traditional schemes when the re-order buffers at the receivers are full.

3. MULTI-PATH TRAFFIC SPLITTING

From the set of non-intersecting paths for each commodity, we need to determine the amount of flow of each commodity across the paths that minimizes congestion. Then, we can assign probability values for each path of every commodity, based on the traffic flow across that path for the commodity. At run time, we can choose the path for each packet from the set of paths based on the probability values assigned to them.

To achieve this traffic splitting, we use a *Linear Programming (LP)* based method to solve the corresponding multi-commodity flow problem. The objective of the LP is to minimize the maximum traffic on each link of the NoC topology, satisfying the bandwidth constraints on the links and routing the traffic of all the commodities in the NoC. Our LP is represented by the following set of equations:

$$\text{min:} \quad t \tag{1}$$

$$\text{s.t} \sum_{\forall j \in 1 \cdots |SP_i|} f_i^j = r_i, \quad \forall i \tag{2}$$

$$\sum_{\forall i} \sum_{\forall j, e_l \in P_i^j} f_i^j = flow_{e_l} \quad \forall e_l \tag{3}$$

$$flow_{e_l} \leq bandwidth_{e_l} \quad \forall e_l \tag{4}$$

$$flow_{e_l} \leq t \ \forall e_l \in P_i^j, \quad \forall i, j \tag{5}$$

$$f_i^j \geq 0 \tag{6}$$

In the objective function we use the variable t to represent the maximum flow on any link in the NoC (refer Equations 1, 5). Equation 2 represents the constraint that the NoC has to satisfy for the traffic flow of each commodity, with the variable f_i^j representing the traffic flow on the path P_i^j of commodity i. The flow on each link of the NoC and the bandwidth constraints are represented by Equations 3 and 4.

Other objectives (such as minimizing the sum of traffic flow on the links) and constraints (like latency constraints for each commodity) can also be used in the LP.

As an example, the latency constraints for each commodity can be represented by the following equation:

$$\sum_{\forall j \in 1 \cdots |SP_i|} (f_i^j \times l^j) \ / \sum_{\forall j \in 1 \cdots |SP_i|} f_i^j \ \leq \ d_i \tag{7}$$

where d_i is the hop delay constraint for commodity i and l^j is the hop delay of path j. Once the flows on each path of a commodity are obtained, we can order or assign probability values to the paths based on the corresponding flows.

4. ADDING FAULT-TOLERANCE SUPPORT

The errors that occur on the NoC links can be broadly classified into two categories: transient and permanent errors. To recover from transient errors, error detection or correction schemes can be utilized in the on-chip network [13]. Forward error correcting codes such as Hamming codes can be used to correct single-bit errors at the receiving NI. However, the area-power overhead of the encoders, decoders and control wires for such error correcting schemes increases rapidly with the number of bit errors to be corrected. In practice, it is infeasible to apply forward error correcting codes to correct multi-bit errors [13]. To recover from such multi-bit errors, switch-to-switch (link-level) or end-to-end error detection and retransmission of data can be performed. This is applicable to normal data packets. However, control packets such as interrupts carry critical information and need to meet real-time requirements. Using retransmission mechanisms can have significant latency penalty that would be unacceptable to meet the real-time requirements of critical packets. Error resiliency for such critical packets can be achieved by sending multiple copies of the packets across one or more paths. At the receiving switch/NI, the error detection circuitry can check the packets for errors and can accept an error free packet. When sending multiple copies of a packet, it is important to achieve the required reliability level for packet delivery with minimum data replication. We formulate the mathematical models for the reliability constraints and consider them in the LP formulation presented in previous section, as follows:

DEFINITION 3. *Let the transient Bit-Error Rate (BER) encountered in crossing a path with maximum number of hops in the NoC be β_t. Let the bit-width of the link (equal to the flit-width) be W.*

We assume a single-bit error correcting Hamming code is used to recover from single-bit errors in the critical packets and packet duplication is used to recover from multi-bit errors. The probability of having two or more errors in a flit received at the receiving NI is given by:

$$\text{P}(\geq 2 \text{ errors}) = \gamma_t = \sum_{k=2}^{W} C_k^W \times \beta_t^k \times (1-\beta_t)^{W-k} \quad (8)$$

When a flit is transmitted n_t times, the probability of having two or more errors in all the flits is given by:

$$\theta_t = \gamma_t^{n_t} \quad (9)$$

As in earlier works ([11]-[13]), we assume that an undetected or uncorrected error causes the entire system to crash. The objective is to make sure that the packets received at the destination have a very low probability of undetected/uncorrected errors, ensuring the system operates for a pre-determined *Mean Time To Failure (MTTF)* of few years. The acceptable residual flit error-rate, defined as the probability of one or more errors on a flit that can be undetected by the receiver, is given by the following equation:

$$Err_{res} = T_{cycle}/(MTTF \times N_c \times inj) \quad (10)$$

where T_{cycle} is the cycle time of the NoC, N_c is the number of cores in the system and inj is the average flit injection rate per core. Each critical packet should be duplicated as many times as necessary to make the θ_t value to be greater than the Err_{res} value, i.e.:

$$\begin{aligned} \theta_t = \gamma_t^{n_t} &\geq Err_{res} \\ \text{i.e. } n_t &\geq ln(Err_{res})/ln(\gamma_t) \end{aligned}$$

The minimum number of times the critical packets should be replicated to satisfy the reliability constraints is given by:

$$n_t = \lceil ln(Err_{res})/ln(\gamma_t) \rceil \quad (11)$$

To consider the replication mechanism in the LP, the traffic rates of the critical commodities are multiplied by n_t and Equation 2 is modified for such commodities as follows:

$$\sum_{\forall j \in 1 \cdots |SP_i|} f_i^j = n_t \times r_i \quad \forall i, \text{critical} \quad (12)$$

To recover from permanent link failures, packets need to be sent across multiple non-intersecting paths. The non-intersecting nature of the paths makes sure that a link failure on one path does not affect the packets that are transmitted on the other paths. The probability of a path failure and the number of permanent path for each commodity (denoted by n_p) can be obtained similar to the derivation of n_t.

Let the total number of paths for a commodity i be denoted by $n_{tot,i}$. Once the number of possible path failures is obtained, we have to model the system such that for each commodity, any set of $(n_{tot,i} - n_p)$ paths should be able to support the traffic demands of the commodity. Thus, even when n_p paths fail, the set of other paths would be able to handle the traffic demands of the commodity and proper system operation would be ensured. We add a set of $n_{tot,i}!/(n_p! \times (n_{tot,i} - n_p)!)$ linear constraints in place of Equation 2 for each commodity in the LP, with each constraint representing the fact that the traffic on $(n_{tot,i} - n_p)$ paths can handle the traffic demands of the commodity.

Thus the paths of each commodity can support the failure of n_p paths for the commodity, provided more than n_p paths exist. When we introduce these additional linear constraints, the impact on the run-time of the LP is small (for our experiments, we did not observe any noticeable delay in the run-time). This is due to the fact that the number of paths available for each commodity is usually small (less than 4 or 5) and hence only few tens of additional constraints are introduced for each commodity.

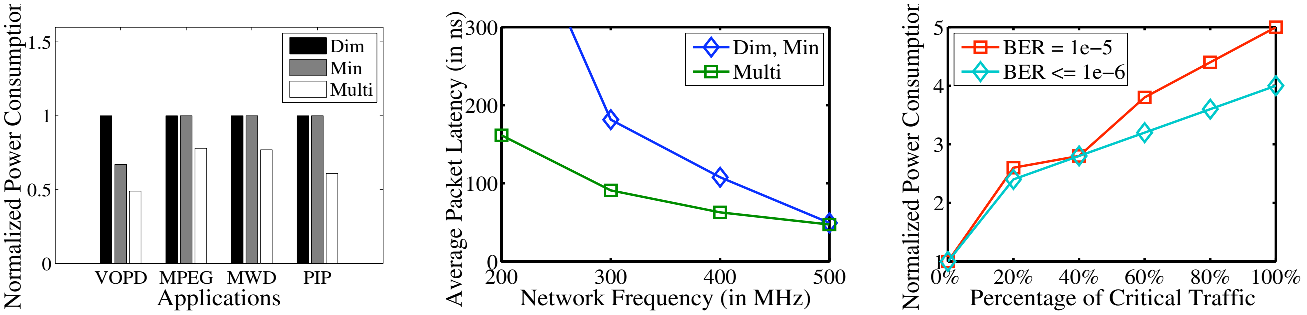
5. SIMULATION RESULTS

The estimated power overhead (based on gate count and synthesis results for switches/NIs from [4]) at the switches/NIs to support the multi-path routing scheme for a 4 × 3 mesh network is found to be 18.09 mW, which is around 5% of the base NoC power consumption. For the power estimation, without loss of generality, we assume that 8 bits are used for representing the source and destination addresses and 8-bit packet identifiers are utilized. The power overhead accounts for the look-up tables and the combinational logic associated with multi-path routing scheme. The numbers assume a 500 MHz operating frequency for the network. The estimated area overhead (from gate and memory cell count) for the multi-path routing scheme is low (less than 5 % of the NoC component area). The maximum possible frequency estimate of the switch design with support for the multi-path routing tables is above 500 MHz, with synthesis results based on the architecture from [4].

5.1 Comparisons with Single-Path Routing

The network power consumption for the various routing schemes: *dimension ordered (Dim)*, *minimum path (Min)* and our proposed *multi-path (Multi)* strategy for different applications is presented in Figure 2(a). The numbers are normalized with respect to the power consumption of dimension-ordered routing. We use several benchmark applications for comparison: *Video Object Plane Decoder (VOPD - mapped onto 12 cores)*, *MPEG decoder (MPEG - 12 cores)*, *Multi-Window Display application (MWD - 12 cores)* and *Picture-in-Picture (PIP - 8 cores) application*. By using the proposed routing scheme, on average we obtain 33.5% and 27.52% power savings compared to the *dimension ordered* and *minimum path routing*, respectively.

The average packet latencies incurred for the MPEG NoC for the different routing schemes is presented in Figure 2(b). The multi-



(a) Routing effects on applications

(b) Average latency for MPEG NoC

(c) Effect of error resiliency

Figure 2: (a) Performance of routing schemes for MPEG. (b), (c) Effect of routing and fault-tolerance on NoC power consumption

path routing strategy results in reduced frequency requirements to achieve the same latency as the single-path schemes for a large part of the design space.

When compared to the multi-path routing scheme with re-order buffers (10 packet buffers/receiver), the current scheme results in 28.25% reduction in network power consumption. The total run time for applying our methodology (includes the run time for path selection algorithms for all commodities and for solving the resulting LP) is less than few minutes for all the benchmarks, when running on a Sun workstation at 1 GHz.

5.2 Effect of Fault-Tolerance Support

The amount of power overhead incurred in achieving fault-tolerance against temporary errors depends on the transient bit-error rate (β_t) of each link and the amount of data that is critical and needs replication. The effect of both factors on power consumption for the MPEG decoder NoC is presented in Figure 2(c). The power consumption numbers are normalized with respect to the base NoC power consumption (when no fault-tolerance support is provided). As the amount of critical traffic increases, the power overhead of packet replication is significant. Also, as the bit-error rate of the NoC increases (higher BER value in the figure, which imply a higher probability of bit-errors happening in the NoC), the amount of power overhead increases. We found that for all BER values lower than or equal to 1e-6, having a single duplicate for each packet was sufficient to provide the required MTTF of 5 years. Adding support for resiliency against a single-path permanent failure for each commodity of the MPEG NoC resulted in a 2.33× increase in power consumption of the base NoC.

6. ACKNOWLEDGMENTS

This work is supported by the US National Science Foundation (NSF, contract CCR-0305718) for Stanford University. It is partially supported by the the Swiss National Science Foundation (FNS, Grant 20021-109450/1), Spanish Government Research Grants TIC 2002/0750 and TIN2005-5619, a Mobility Post-Doc Grant from UCM for David Atienza, by the Semiconductor Research Corporation (SRC, project number 1188) and a grant by STMicroelectronics for University of Bologna.

7. CONCLUSIONS

The re-order buffers required in traditional multi-path schemes have large area, power overhead and deterministically sizing them is infeasible in practice. In this work, we have presented a multi-path routing strategy that guarantees in-order packet delivery at the receiver. We introduced a methodology to split the application traffic across the multiple paths to obtain a network operation with minimum power consumption. We also integrated with the methodology, the use of spatial and temporal redundancy to tolerate transient as well as permanent errors occurring on the NoC links. Our method results in large NoC power savings for several SoC designs when compared to traditional single-path systems.

8. REFERENCES

[1] L.Benini and G.De Micheli, "Networks on Chips: A New SoC Paradigm", IEEE Computers, pp. 70-78, Jan. 2002.

[2] D.Wingard,"MicroNetwork-Based Integration for SoCs", Design Automation Conference DAC 2001, pp. 673-677, Jun 2001.

[3] P.Guerrier, A.Greiner,"A generic architecture for on-chip packet switched interconnections", DATE 2000, pp. 250-256, March 2000.

[4] S. Stergiou et al., "×pipesLite: a Synthesis Oriented Design Library for Networks on Chips", pp. 1188-1193, Proc. DATE 2005.

[5] J. Hu, R. Marculescu, 'Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures', Proc. DATE, March 2003.

[6] S. Murali, G. De Micheli, "SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs", Proc. DAC 2004.

[7] S. Murali, G. De Micheli, "Bandwidth Constrained Mapping of Cores onto NoC Architectures", Vol. 2, pp. 20896-20899, DATE 2004.

[8] A. Hansson et al., "A unified approach to constrained mapping and routing on network-on-chip architectures", pp. 75-80, Proc. ISSS 2005.

[9] J. Kim et al., "A low latency router supporting adaptivity for on-chip interconnects", Proc. DAC, June 2005.

[10] J. Hu, R. Marculescu, 'DyAD - Smart Routing for Networks-on-Chip', Proc. DAC, June 2004.

[11] R. Hegde, N. R. Shanbhag. "Towards Achieving Energy Efficiency in Presence of Deep Submicron Noise". IEEE
Trans. on VLSI Systems, 8(4):379-391, August 2000.

[12] D. Bertozzi et al., "Error Control Schemes for On-Chip Communication Links: The Energy-Reliability Trade-off", IEEE Trans. on CAD, Vol. 24, No. 6, pp. 818-831, June 2005.

[13] S. Murali et al., "Analysis of Error Recovery Schemes for Networks on Chips", IEEE Design and Test of Computers, Vol. 22, No. 5, pp. 434-442, Sep/Oct 2005.

[14] R. Marculescu, "Networks-On-Chip: The Quest for On-Chip Fault-Tolerant Communication", Proc. of IEEE ISVLSI, 2003.

[15] H. Zimmer et al.,"A Fault Model Notation and Error-Control Scheme for switch-to-Switch Buses in a Network-on-Chip", ISSS/CODES, 2003.

[16] F. Worm et al., "An Adaptive Low-power Transmission Scheme for On-chip Networks" , ISSS, October 2002, pp. 92-100

[17] M. Pirretti et al.,"Fault Tolerant Algorithms for Network-On-Chip Interconnect", Proc. of ISVLSI, Feb 2004.

[18] S. Manolache et al., "Fault and Energy-Aware Communication Mapping with Guaranteed for Applications Implemented on NoC", Proc. DAC 2005.

[19] W. J. Dally, B. Towles, "Principles and Practices of Interconnection Networks", Morgan Kaufmann , Dec 2003.

[20] W. J. Dally et al., "The Avici terabit switch/router", Proc. Hot Interconnects, Aug. 1998.

[21] Graig B. Stunkel, et al.; "The SP2 Communication Subsystem, " IBM Technical Report, August 22, 1994.

[22] Y. Aydogan et al. , "Adaptive Source Routing in Multistage Interconnection Networks", Proc. International Parallel Processing Symposium, 1996.